

AD-A168 403

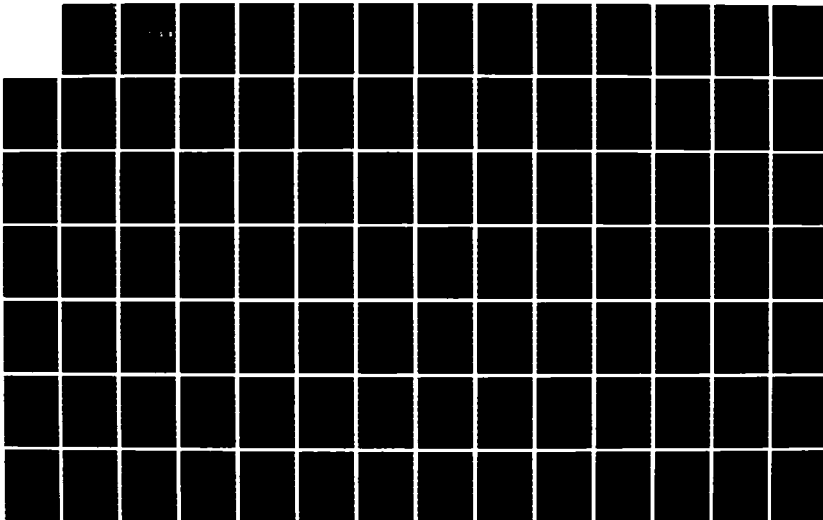
RECOGNITION OF TWO-DIMENSIONAL SHAPES(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA G P QUEK MAR 86

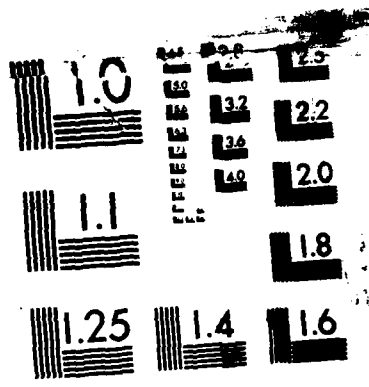
1/8,

UNCLASSIFIED

F/G 9/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A168 403



DTIC
ELECTE
JUN 09 1986
S D

THESIS

RECOGNITION OF TWO-DIMENSIONAL SHAPES

by

Quek Gim Pew
March 1986

Thesis Advisor:

C. H. Lee

Approved for public release; distribution is unlimited

DTIC FILE COPY

86 6 9 082

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b DECLASSIFICATION / DOWNGRADING SCHEDULE			
4 PERFORMING ORGANIZATION REPORT NUMBER(S)		5 MONITORING ORGANIZATION REPORT NUMBER(S)	
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (if applicable) Code 62	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	
8a NAME OF FUNDING / SPONSORING ORGANIZATION	8b OFFICE SYMBOL (if applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code)		10 SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO	PROJECT NO
		TASK NO	WORK UNIT ACCESSION NO
11 TITLE (Include Security Classification) RECOGNITION OF TWO-DIMENSIONAL SHAPES			
12 PERSONAL AUTHOR(S) G. P. Quek			
13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM TO	14 DATE OF REPORT (Year, Month, Day) 1986 March	15 PAGE COUNT 103
16 SUPPLEMENTARY NOTATION			
17 COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		Shape Recognition, Partial Occlusion, Scale and Orientation Invariance, Correlation	
19 ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>A new scheme for coding the boundary of two-dimensional shapes is proposed. Random points on the boundary are paired for this coding. Using this scheme, an effective and efficient correlation technique to match two-dimensional shapes is developed.</p> <p>This technique has a number of very desirable characteristics. It is able to match shapes of arbitrary scale and orientation. The given shape may have closed or open boundary or even have portion of it obstructed from view. Matching can be performed at varying degrees of details, giving this technique an added robustness against geometric distortions. It also has the capability to discriminate between different shapes.</p> <p>Computation time on the IBM 3033 computer is typically 10 CPU seconds to generate one correlation curve between two shapes, each with a 500-point boundary curve.</p>			
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL C. H. Lee		22b TELEPHONE (Include Area Code) (408) 646 2452	22c OFFICE SYMBOL 62Le

Approved for public release; distribution is unlimited.

Recognition of Two-Dimensional Shapes

by

Quek Gim Pew
Ministry of Defence, Singapore
B. Eng. (Hons), National University of Singapore 1981

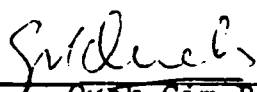
Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

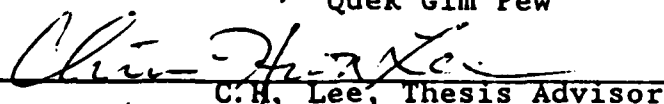
from the

NAVAL POSTGRADUATE SCHOOL
March 1986

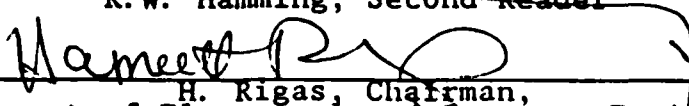
Author:

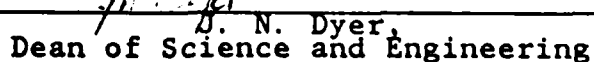

Quek Gim Pew

Approved by:


C.H. Lee, Thesis Advisor


R.W. Hamming, Second Reader


H. Rigas, Chairman,
Department of Electrical and Computer Engineering


N. Dyer,
Dean of Science and Engineering

ABSTRACT

A new scheme for coding the boundary of two-dimensional shapes is proposed. Random points on the boundary are paired for this coding. Using this scheme, an effective and efficient correlation technique to match two-dimensional shapes is developed.

This technique has a number of very desirable characteristics. It is able to match shapes of arbitrary scale and orientation. The given shape may have closed or open boundary or even have portion of it obstructed from view. Matching can be performed at varying degrees of details, giving this technique an added robustness against geometric distortions. It also has the capability to discriminate between different shapes.

Computation time on the IBM 3033 computer is typically 10 CPU seconds to generate one correlation curve between two shapes, each with a 500-point boundary curve.



Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION	10
II.	SURVEY	14
	A. INTRODUCTION	14
	B. TEMPLATE MATCHING	14
	C. FEATURE MATCHING	16
	D. TRANSFORM PARAMETER MATCHING	19
	E. CONCLUSION	23
III.	PRELIMINARY FINDINGS	24
	A. IDEAL SHAPE REPRESENTATION	24
	B. DIFFICULTIES IN REPRESENTATION	25
	C. DIFFICULTIES IN MATCHING	25
	D. SCALE AND ORIENTATION INVARIANT REPRESENTATION	31
	E. SCALE AND ORIENTATION INVARIANT HOUGH TRANSFORM	37
IV.	A NEW CORRELATION TECHNIQUE	44
	A. INTRODUCTION	44
	B. RANDOM CODING	44
	C. FEATURES	46
	1. Scale and Orientation Invariance	46
	2. Robustness	47
	3. "Local" Characteristics	48
	4. Discrimination	48
	5. "End Losses"	49
	6. Lack of Internal Consistency Checks	51
	D. RESULTS	51
	1. Geometric Distortion	51
	2. Partial Matching	62

3. Discrimination Capability	84
E. CONCLUSIONS	95
V. SUMMARY	96
APPENDIX: GENERATION OF TEST SHAPES	98
LIST OF REFERENCES	100
INITIAL DISTRIBUTION LIST	102

LIST OF TABLES

1.	R-TABLE	22
2.	R-TABLE FOR $\beta - \Phi$ CODING	40

LIST OF FIGURES

1.1	Two Shapes to be Matched	10
1.2	A Sample Shape to be Described	12
2.1	Hough Transform	21
3.1	Hypothetical Boundary Representations	28
3.2	Correspondence Chart - Raw Data	29
3.3	Correspondence Chart - Processed Data	30
3.4	Arc to Chord Length Ratio Illustration	33
3.5	β -s Representation for R35-52 and R34-3lp	34
3.6	β -s Representation for R35-52 and R34-102	35
3.7	Two Other Possible Specifications Besides ACR	36
3.8	β - ϕ Coding	39
3.9	Matching of R35-52 and R34-3lp Using β - ϕ Correlation with ACR Specification	41
3.10	Matching of R35-52 and R34-102 Using β - ϕ Correlation with ACR Specification	42
4.1	β - ϕ Coding Using Random Separation	46
4.2	Matching in the Presence of Noise	48
4.3	Discrimination Using Secondary Specifications	50
4.4	Correlation Between R35-52 and R35-52	52
4.5	Correlation Between R35-52 and R34-102	53
4.6	Correlation Between R35-52 and R34-152	54
4.7	Correlation Between R35-52 and R34-252	55
4.8	Correlation Between R35-52 and R35-52 Using a Wider Search Range	57
4.9	Correlation Between R35-52 and R32-011r	58
4.10	Correlation Between R35-52 and R32-11r	59
4.11	Correlation Between R35-52 and R32-31r	60
4.12	Correlation Between R35-52 and R32-51r	61
4.13	Correlation Between R35-52 and R32-51r Using Relaxed Tolerances	63
4.14	Correlation Between R44-52 and R43-31	64

4.15	Correlation Between R14-52 and R14-92	65
4.16	Correlation Between R14-52 and R14-51	66
4.17	Correlation Between R14-52 and R13-01ln	67
4.18	Correlation Between R25-52 and R24-92	68
4.19	Correlation Between R25-52 and R23-1ln	69
4.20	Correlation Between R25-52 and R24-202	70
4.21	Correlation Between R35-52 and R34-3lp	71
4.22	Detailed Correlation Between R35-52 and R34-3lp for Coding Range 150 to 200	72
4.23	Matched Segment Between R35-52 and R34-3lp at Zero Degree Relative Orientation	74
4.24	Correlation Between R32-01lr and R34-22p	75
4.25	Correlation Between R32-1lr and R34-22p	76
4.26	Correlation Between R32-3lr and R34-22p	77
4.27	Matched Segments Between R32-3lr and R34-22p at -75 Degrees Relative Orientation	78
4.28	Correlation Between R32-3lr and R33-22p	79
4.29	Correlation Between R33-22p and R32-1lr	80
4.30	Matched Segments Between R33-22p and R32-1lr at 95 Degrees Relative Orientation	81
4.31	Correlation Between R43-2lp and R44-52	82
4.32	Correlation Between R14-52 and R13-05lp	83
4.33	Correlation Between R35-52 and R25-52	85
4.34	Correlation Between R35-52 and R14-52	86
4.35	Correlation Between R35-52 and R44-52	87
4.36	Correlation Between R44-52 and R25-52	88
4.37	Correlation Between R14-52 and R25-52	89
4.38	Correlation Between R14-52 and R44-52	90
4.39	Correlation Between R33-22p and R44-52	91
4.40	Matched Segments Between R33-22p and R44-52 at -175 Degrees Relative Orientation	92
4.41	Correlation Between E3-152 and E2-031	93
4.42	Correlation Between E3-152 and E3-22	94

ACKNOWLEDGEMENTS

The author would like to thank his advisor, Prof. C. H. Lee and second reader, Prof. R. W. Hamming for the guidance and numerous discussions that help towards formulating the ideas in this thesis.

I. INTRODUCTION

This thesis investigates the following problem. Given the outlines of two objects, determine whether there are any regions where they have the same shape. It is implicit that one of objects may be partially occluded so that only a portion of it is available. Minimum restriction is placed on the class of objects to be matched. The objects may have closed or open boundaries (e.g., images of coastlines), with arbitrary scale and orientation. Furthermore, the matching must be done in the presence of noise (i.e., geometric distortions). An example of the type of shapes that will be studied in this report is given in Figure 1.1.

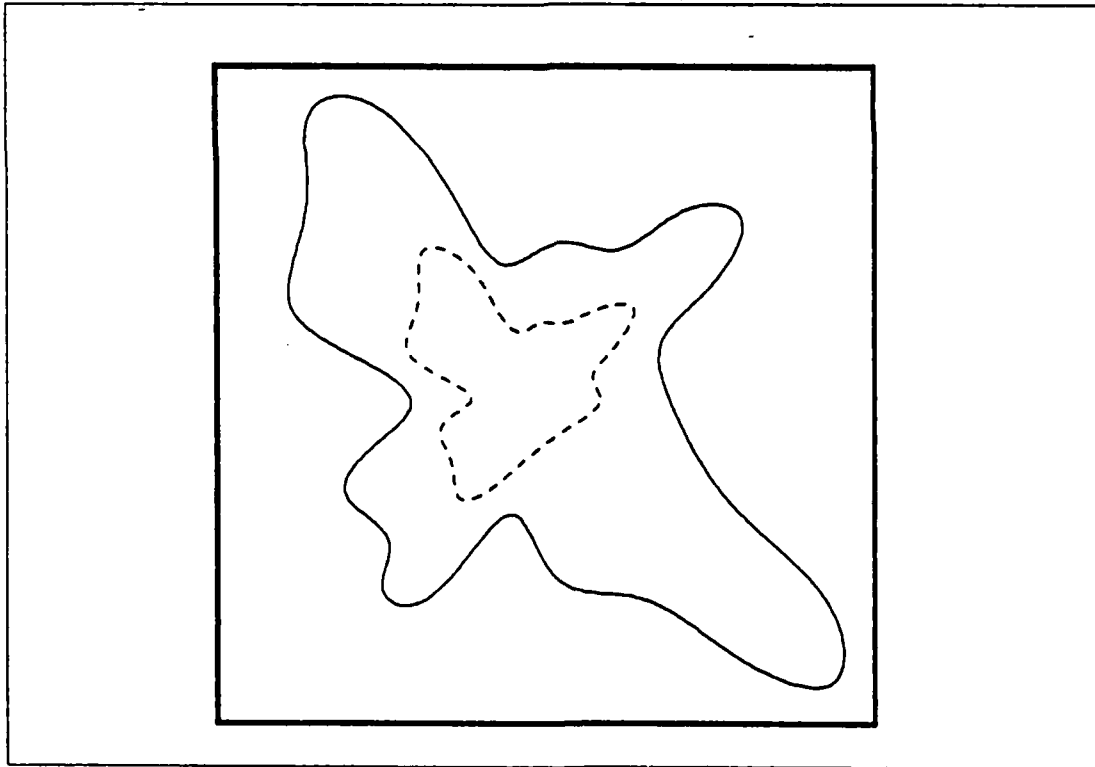


Figure 1.1 Two Shapes to be Matched

The shape of an object contains a great deal of information of the object. This is evident from our ability to recognise or at least guess at objects from their shapes alone. It is thus not surprising that the problem of shape description and recognition is fundamental in computer vision.

Shape is, unfortunately, a largely qualitative concept. Although we possess intuitive ability for dealing with shape, we lack a good quantitative description. Shape is apparently implicit in our language, where the name of an object itself contains its shape structure. To appreciate this, consider Figure 1.2 (adapted from Freeman [Ref. 1]). Suppose one is required to convey this figure to a distant friend, say over the telephone. How would one proceed? One could possibly spend a long time describing it in terms of the 'two peaks', 'left gentle slope', 'right steep cliff', etc and yet at the end of it, still doubtful whether the message is brought across. Consider the alternative description of 'steep forehead, medium-sized nose, thin lips and a prominent chin'! (This is of course not just restricted to our perception of shape. We have the same difficulty with some of the other sensory perceptions too. Thus we speak of 'lemony taste' and 'silky smoothness'.) The main problem in programming a machine to deal with shape lies largely in the need to 'explicitize' shape.

Researchers in this field have lamented that there is little guidance from the traditional mathematics [Ref. 2: p. 229]. As pointed out by Blum [Ref. 3], geometry has its roots in surveying and has developed closely along with the physical sciences. The general Cartesian view of geometry metrizes a space and describes a curve in that metric in some functional form. He observed that this constrained analysis to shapes of simple functional form rather than geometric structure.

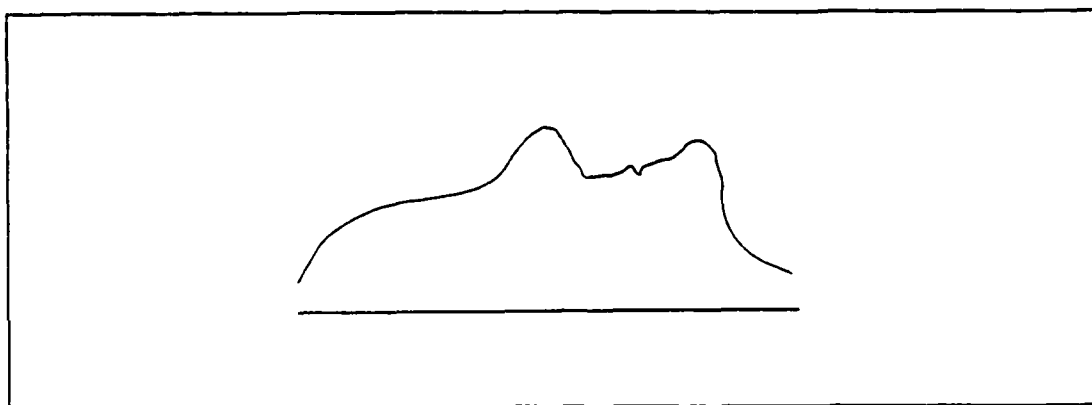


Figure 1.2 A Sample Shape to be Described

There has been extensive research on the subject of shape representation and recognition [Ref. 4]. Many ad hoc techniques have been developed, so that a large assortment of tools is now available for solving certain practical problems. And, as noted by Rosenfeld in his review paper [Ref. 5], the field has begun to develop a scientific basis. Recent developments in representation structures in mathematics have also allowed researchers to move away from the traditional framework of vector space (using classical mathematical tools of analysis and linear space) to that of a structural framework (using modern tools such as graphs and grammars).

Applications of computer vision are wide and varied. These include character recognition, fingerprint identification, microscopy, radiology, robot vision, remote sensing and navigation, to name a few. Many of the successful application of shape recognition have been primarily two-dimensional. The most general problem of recognition of a partially occluded three-dimensional object of unknown scale, orientation and aspect remains a research topic.

This thesis is confined to two-dimensional shapes. It assumes that the outline of the object has been extracted and pre-processed to smoothen out some of the noise. Early

in this investigation, it was realised that our problem is two-fold. There is the representation problem and the matching problem (recognition and matching will be used interchangeably throughout this report). The representation problem is largely geometric in nature, whereas matching is primarily an algorithmic problem. However, the means of representation determines the complexity of the matching algorithm, and more importantly, it places a limit on the capability of the matching algorithm. Thus, a representation based on Fourier Descriptors, for example, would not be able to handle the partial occlusion problem because of its global nature.

The following chapter contains a survey of the various techniques that have been developed for the analysis of two-dimensional shapes. Chapter Three summarizes the initial findings of this investigation and introduces a new representation and matching algorithm. This representation scheme is both scale and orientation invariant. The matching algorithm is similar to the Hough Transform, but it has several distinct features that make it scale and orientation invariant too. Chapter Four presents the final results of this investigation - a new correlation technique that is simple and robust. This technique is applied to a number of test shapes and the results verify that it is capable of recognising parts of a shape. The shape may be of unknown scale and orientation. The ability to discriminate two different shapes is also demonstrated. The weakness of this technique is also discussed. Finally, the last chapter summarizes the key results obtained and offers suggestions for future work.

II. SURVEY

A. INTRODUCTION

The recognition of shape is a relatively old problem that has been recently taken up by engineers and computer scientists. Psychologists have long puzzled over the ability of humans and animals to discriminate shapes. A collection of very interesting papers on the early studies on form perception and discovery can be found in Uhr [Ref. 6]. The early experiments conducted had suggested that the information in an object outline is concentrated at those points having high curvature. This idea is in fact the basis for several of the current techniques for shape recognition [Ref. 7: p. 165].

This chapter contains a survey of the techniques developed for two-dimensional shape recognition. It is not intended to be a complete survey, but rather to be indicative of the variety of techniques that have been examined and also to demonstrate the difficulties facing researchers in this area.

For convenience, these techniques are grouped into three categories, according to the matching scheme used. These are

- a. Template matching
- b. Feature matching
- c. Transform parameter matching

B. TEMPLATE MATCHING

Template matching is the oldest technique developed. This is basically a two-dimensional cross-correlation between the reference shape (the 'template') and the test shape. One may visualize template matching by imagining the template being shifted across the test shape to different

offsets and determining the amount of overlap. In its basic form, template matching is of limited use.

Many variants to this basic method have been proposed. Most of these involve some sort of hierarchical template matching process. In this, sub-templates for parts of the objects are first matched. One then looks for combination of partial matches in approximately the correct relative positions. The computation cost is obviously high. Also, template matching breaks down when the two shapes to be matched are of different scales.

The two-dimensional correlation can be converted to a one-dimensional correlation by coding the boundary in some appropriate functional form. Possible coding schemes include radius-angle representation, orientation-arc length representation, curvature-arc length representation.

The radius-angle (or polar) representation requires a reference origin. This is usually taken to be the object's centroid. This representation is obviously scale-dependent. The need for a reference origin also makes it unsuitable for partially occluded objects and those with open boundaries. Also the need for the representation to be single-valued further restricts the type of shapes that can be coded in this manner.

The orientation-arc length representation codes the angle made between a fixed axis and a tangent to the boundary as a function of the arc length. This representation is scale invariant, but not orientation invariant. Straight horizontal lines in this representation correspond to zero curvature (ie. straight lines in the boundary), and straight non-horizontal lines correspond to segments of circle with the radii of curvature given by the slopes of the lines. (This allows the boundary to be easily segmented into straight lines and circular arcs and is used sometimes in the initial processing for feature matching).

The curvature-arc length representation codes the curvature of the boundary as a function of arc length. This representation is orientation invariant. Unfortunately it is not scale independent. (A circle of radius r , for example, has a curvature of $1/r$). Also, curvature is very sensitive to noise. Curvature is, however, a popular descriptor and this representation is often used to extract the extremas (in curvature) for feature matching [Ref. 8].

A discrete version of the orientation-arc length representation has also been used. Commonly called the chain codes, this codes the boundary into short line segments that lie on a fixed grids with a fixed set of orientation. Although efficient in representation and cross-matching, chain codes are rather sensitive to noise and have other shortcomings that made this representation unsuitable for general shape matching. [Ref. 9]

None of the representation discussed above is simultaneously scale and orientation invariant. The problems in obtaining a 'truly intrinsic' representation of the boundary is further discussed in the next chapter.

C. FEATURE MATCHING

Another approach to shape matching is to construct a structural model of the shape. This model describes the spatial decomposition of a shape in terms of features or shape primitives. There are no established guidelines for choosing shape primitives; however it is desirable that these primitives provide a compact description of the shape and be easily extracted from the shape.

A reading through the literature reveals a wide variety of primitives that have been used. Most of these are based (explicitly or implicitly) on curvature. These include curvature maxima and minima, corners, protrusions, intrusion, linear segments, quadratic segments, circular arcs, convex blobs, T-shaped parts, etc. (see for example [Refs. 10,11])

These primitives are often further qualified by a set of attributes, e.g., large, sharp convex corner facing North. Once the primitives are obtained, relationships between them are computed. Examples of these relationships are adjacency, collinearity, symmetry, etc.

The matching algorithm depends on the type of structural model. There are essentially two kinds of structural models, the relational model and grammatical model [Ref. 12: pp. 426 to 434]. In relational model, the primitives appear as nodes in a tree or graph structure. Nodes are connected by their relationship. The matching algorithm typically involves a search for correspondence nodes in the two relational structures to be matched.

Grammatical model makes use of formal language theory to describe how the primitive pieces of the shape are joined together. A grammar consists of three types of entities: terminal or primitive symbols, non-terminal symbols and production rules. A grammar can be used to construct strings of primitive symbols (called a sentence) by successive application of the production rules. The set of all sentences that can be generated using a given grammar is called the language of the grammar. Object recognition is then a process of determining whether a sentence (which describes the object) belongs to a given language, by parsing it with respect to the grammar of the language.

A major problem with the grammatical model is the construction of a grammar that is comprehensive enough to generate all the possible types of shapes of interest and yet discriminatory enough to reject others. A number of grammars have been developed over the years. A good description of these can be found in [Ref. 13: pp. 365 to 382].

A common problem with these relational and grammatical models is the effect of noise. Noise complicates the process of computing the appropriate structures. This is

normally handled by preprocessing the shape boundary, usually by some sort of piecewise linear fit (polygonal approximation) [Ref. 14: p. 275]. Here one runs into the problem of how to locate the breakpoints, ie. when should a linear segment ends and a new segment begins [Ref. 2: p. 232]. A number of criteria have been proposed [Ref. 7: pp. 168 to 184]. Recently the use of piecewise polynomial (mainly B-splines) has become popular. B-splines have a number of computational and representation advantages. For example, its 'local' characteristics and 'terse' representation allow programs to manipulate them easily [Ref. 2: p. 239]. As with piecewise linear approximation, B-splines approximation is also sensitive to the placement of breakpoints (knots).

It is evident that within the structural framework, one gains a considerably greater representation freedom, but loses the convenience of vector space and the analytical tools there. The shape primitives and their relationships tend to be more qualitative than quantitative in description. For example, a primitive like 'sharp corner' does not carry numerical values of the degree of sharpness or the extent of the corner. Without a quantitative description, standard similarity measures such as least mean square differences cannot be easily applied. This also implies that the feature matching technique performs better in classifying shapes into their generic classes (those generated by the particular grammar) than in distinguishing between objects from the same class.

This approach is highly suited for scene understanding application where a 'literal', ie. qualitative, description of the scene can be built up and compared with another scene [Ref. 5]. It is of limited use in applications such as change detection, where detailed matching of specific boundaries is required. This technique will not be further discussed in this report.

D. TRANSFORM PARAMETER MATCHING

The above two classes of matching techniques operate on the original two-dimensional spatial information. Another approach is to transform the original data into a different domain and to perform the matching in this new domain. This method is no doubt motivated by the success of the frequency approach in electrical engineering analysis. It is thus not surprising that the Fourier series representation of the parameterized boundary is one of the oldest and most popular transform technique.

The boundary may be coded in any of the representation schemes discussed in the earlier section. These representations are periodic, and can thus be expanded into a Fourier series. A common feature of the Fourier Descriptors (as these coefficients of the series are called) is that the general shape is given rather well by a few of the low-order terms (important for data compression applications). Properly parametrized, the coefficients can be made independent of scale and orientation [Ref. 2: p. 238].

However this description is global in nature, ie. each coefficient depends on every points on the boundary. It is therefore not suitable for matching partially occluded objects. Also, the Fourier descriptors can distinguish among symmetrical curves only on the basis of the phase of the descriptors. This, unfortunately, cannot be reliably computed in many cases. Thus, the descriptors of the contours of '2' and '5' are virtually identical [Ref. 4].

In contrast to the Fourier descriptors which describe the boundary, another transform technique, the method of moments, describes the shape interior points. In this technique, coordinates of points belonging to the shape are used to compute a set of moments. These moments can be normalised to obtain measures that are invariant under scaling and rotation [Ref. 13: p. 354]. It is difficult to relate higher

moments to the shape, and furthermore, this is also a global transform; thus it is not suitable for partially hidden objects too.

A new transform technique appeared in the literature recently [Ref. 15]. It treats a shape outline as a set of discrete data that is generated by an autoregressive model. An autoregressive model is a parametric equation that expresses each sample of an ordered set of data samples as a linear combination of a specified number of previous samples from the set plus an error term. This model is widely used in speech modelling and spectral estimation. The shape is then described by the model parameters.

However, unlike conventional digital signal processing where the sample interval is determined physically (and uniquely) by an external reference (namely time), the samples obtained from a shape boundary is determined by the scale factor of the image of the object. It can be made scale independent if the samples are taken at fixed angular interval from, say, the centroid of the shape. The centroid is, however, a global feature, which then makes this scheme unsuitable for partially occluded objects.

Another interesting transform technique makes use of geometric transformation to map instances of a given shape pattern into peaks of a transform space. This so-called Hough Transform was originally developed to handle simple shapes such as straight lines and circles, but it was recently extended to arbitrary shapes [Ref. 16]. We will describe this technique in some details as it will be the basis for a new matching algorithm to be developed in the next chapter. The description below is adapted from Ballard [Ref. 2: p. 128].

Consider an object with known scale and orientation. Pick a reference point (x_c, y_c) in the silhouette (see Figure 2.1). At each boundary point (x_i, y_i) , compute the gradient

direction (ϕ_i) and the vector r . The magnitude of this vector is the length of the line joining the reference point to the boundary point and the direction is given by the angle between this line and the x-axis (α). Store r as a function of ϕ_i . This representation is multivalued, and in general an index ϕ_i may have many values of r . The set of all such vectors indexed by ϕ_i forms what is called the R-table. Table 1 shows the form of the R-table.

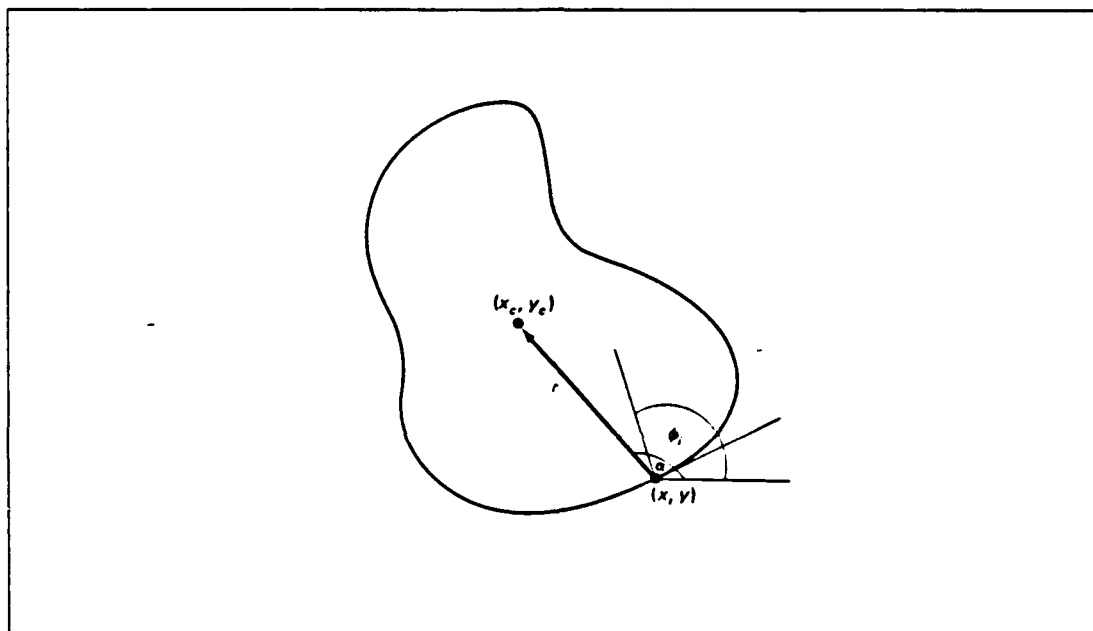


Figure 2.1 Hough Transform

The R-table is used to detect instances of a shape as follows. First, an accumulator array of possible locations of the reference point $A(x_c, y_c)$ is initialised to zero. For each boundary point of the test shape, compute its gradient angle (ϕ_i). For each vector indexed by this angle in the R-table, compute the possible centers of the reference point. That is, for each table entry of ϕ_i , compute

$$\begin{aligned} x_c &= x_i + r(\phi_i) * \cos[\alpha(\phi_i)] \\ y_c &= y_i + r(\phi_i) * \sin[\alpha(\phi_i)] \end{aligned}$$

Next, increment the accumulator array corresponding to this location, ie.,

$$A(x_c, y_c) = A(x_c, y_c) + 1$$

The peaks in the accumulator array then correspond to possible instances of the shape.

TABLE 1
R-TABLE

Angle Measured from Boundary to Reference Point	Set of Vectors $r = (r, \alpha)$
ϕ_1	$r_{11}, r_{12}, \dots, r_{1n}$
ϕ_2	$r_{21}, r_{22}, \dots, r_{2p}$
.	.
.	.
.	.
ϕ_m	$r_{m1}, r_{m2}, \dots, r_{mq}$

This technique can be summarised as follows. For the reference shape, code the boundary with respect to a fixed reference point. For the test shape, use this coding to reconstruct the possible locations of the reference point. A cluster of possible locations would be obtained. If the two shapes are identical, there would be a peak at the location of the original reference point.

In this form, the Hough Transform has several limitations. It requires the reference and test objects to be of the same scale and orientation. Computational complexity increases rapidly if it is necessary to deal with variations in scale and orientation. Thus, to account for orientation,

the above procedures must be repeated for every orientation to be distinguished. If it is required to distinguish orientation, say, 10 degrees apart, the procedures must be repeated 36 times, resulting in 36 accumulator arrays. The best match would then be identified by the accumulator with the largest value in all of the 36 arrays. Similarly with scale variations. A more serious objection is that the transform suffers from false peaks in the accumulator array due to random matches.

In the next chapter, it will be shown how with a different boundary representation scheme, this method can be modified to make it scale and orientation invariant. Chapter Four presents an improved version that also tends to decorrelate these random matches.

E. CONCLUSION

There exists a wide variety of techniques for shape representation and matching. However, each technique has its limitations and is restricted to its specific domain of shapes. The question naturally arises. Is there a scheme of representation and matching that is simultaneously scale and orientation invariant and also capable of handling partially occluded objects? We address this question in the next chapter.

III. PRELIMINARY FINDINGS

A. IDEAL SHAPE REPRESENTATION

The manner in which the shape boundary is represented determines to a large extent the capability and complexity of the matching algorithm. If the representation makes use of global information, then partial matching would not be possible. If the representation is not orientation invariant, then the matching algorithm would have to be repeated across the range of possible orientations.

We can formulate a number of desirable characteristics that the ideal shape representation might possess (see also [Ref. 17]). These are:

- a. It should be local. By this we mean (i) the coding of each point on the boundary is determined by a short section of the boundary, rather than by the entire boundary, and (ii) the coding is not dependent on an external reference, such as a centroid.
- b. It should be independent of the orientation and scale of the shape.
- c. It should be bounded. In other words, a small change to part of the boundary should create a small local change in the representation.
- d. It should allow for efficient and robust matching in the presence of noise (geometric distortion).
- e. It should uniquely specified a single boundary (up to the equivalence classes induced by scaling and rotation).
- f. It should contain information about the boundary at varying levels of detail, so that the matching process could be performed at different levels of coarseness.
- g. It should be easily computable efficiently.

These characteristics are ideal, and it is by no means obvious from the outset, that a representation with such characteristics could be found. Later in this chapter we shall describe one scheme of representation and matching that comes close to satisfying these characteristics.

B. DIFFICULTIES IN REPRESENTATION

For a representation to be scale and orientation invariant, it is necessary that it be local. Unfortunately, this is not a sufficient condition. It is necessary because if an external reference is used this must be related to the boundary, either in distance or direction. This immediately ties the representation to a fixed scale or orientation. That it is not sufficient can be seen from the fact that the curvature-arc length representation is local in nature, and yet is scale dependent. It is not obvious what the sufficient condition(s) is(are). Rather than look for these, the author concentrated on finding local representation that is both scale and rotation invariant.

In a local representation, each point is influenced by a small section of the boundary. The question immediately arises. - How to determine this section? It is obvious that the 'extent' of this section must be determined on a 'local' basis too. This 'extent' cannot be determined by factors such as 'length' or 'number of points' without making it scale dependent.

The difficulties with shape representation can be traced to the basic fact that one cannot associate an absolute external reference with shape, as one could associate, say time, with radar signals. Shape is a spatial variation, and the spatial coordinates are, unfortunately, relative in nature. Radar signals, on other hand, is a temporal variation, and for all practical purposes, time is an absolute coordinate; there is no ambiguity regarding the interval of time and the 'direction' of time.

C. DIFFICULTIES IN MATCHING

The primary problem with matching is our lack of knowledge on how to deal with geometric distortion (noise). Almost all forms of shape representation (boundary and structural codings) are sensitive to geometric distortions.

As mentioned before, most researchers use some form of hierarchial schemes in the matching process. We could, for example, first find matches to small pieces (the smaller the pieces, the less the effect of distortion), then look for consistent combination of these matches. Alternatively, we could first find matches at low resolution (rough details) and then search for higher resolution matches in the vicinity of the lower resolution matches. These hierarchial schemes increase the matching complexity (more so if the representation is not scale and rotation invariant) and the computation cost.

In contrast, conventional signal processing makes extensive use of the statistical properties of the signal and noise in order to extract the signal. In shape recognition, we have very little understanding of the properties of geometric distortion (noise) and how this could be filtered out. There is little or no work done in this area. (It should be added that it is also not obvious how this problem should be attacked). Most researchers concentrated on specific matching algorithm, using for the most parts, ad-hoc methods.

A second, more mundane, problem is concerned with correlation matching. Any representation that uses the arc length as one of the coordinate has to content with the fact that both scale changes and geometric distortions (noise) affect the length of arc traversed during the coding. Thus even though the representation may be scale invariant, (in that the particular characteristics at each boundary point that is been coded does not vary with scale changes), the unknown factor in the arc length axis makes matching using correlation difficult. If the shapes to be matched are complete, then the scale factor could be possibly removed by normalizing with respect to the boundary length.

One simple algorithm to correlate scale and orientation invariant representations at different scaling in the arc length axis was devised. This algorithm basically builds up a diagram of correspondence points of the two curves to be matched. The algorithm is described below.

Algorithm 1: Correlation Matching

- a. Set up an array, $A(i,j)$ of dimension M by N where M, N are the number of points of curve 1 (denoted by $f(n)$) and curve 2 (denoted by $g(n)$). Initialize the array to zeros.
- b. For each point of $f(n)$, search through the points of $g(n)$ for those points that match (to within a specified tolerance). Change the corresponding array entry to 1, ie.,

$$A(i,j) = 1 \quad \text{if} \quad f(i) = g(j)$$

- c. If the array values are plotted (point for '1', blank for '0'), a scatter diagram would result. Linear segments in this diagram correspond to matched segments of the two curves. The slopes and intercepts of these linear segments give the relative scale and orientation of the matched segments of the two boundaries.

An illustration of this can be seen in Figures 3.1, 3.2, 3.3. Figure 3.1 shows the hypothetical boundary representation of two shapes to be matched. It is assumed that these shapes have been coded in a representation scheme that is scale and orientation invariant. The two shapes differ in scale (as can be seen in their arc lengths) and orientation (as evident in the cyclical shift). There is also some distortion over a section of the boundary (points 1 to 60 in $g(n)$). Figure 3.2 shows the 'scatter diagram' or correspondence chart. This is a very busy chart. (It is interesting

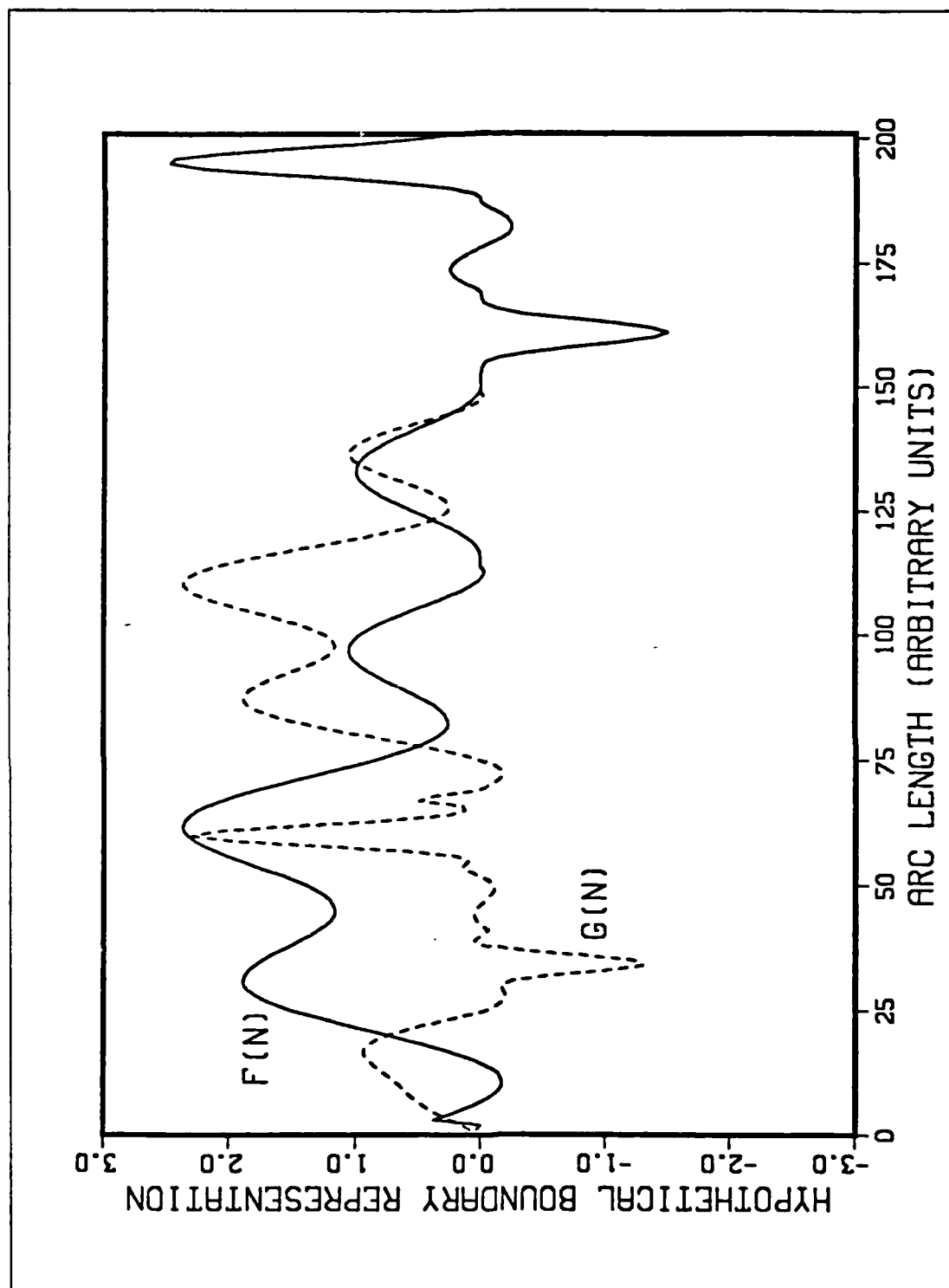


Figure 3.1 Hypothetical Boundary Representations

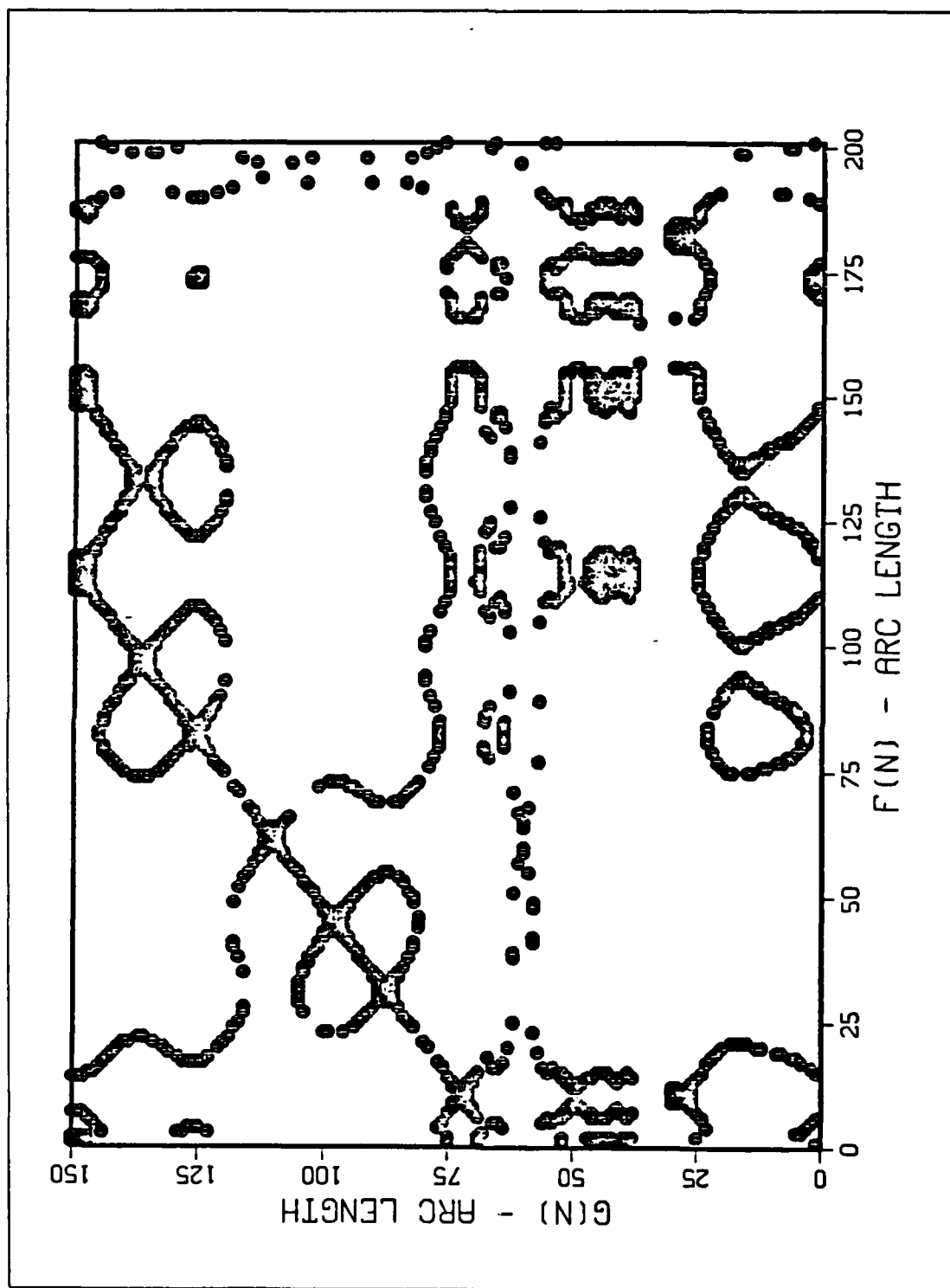


Figure 3.2 Correspondence Chart - Raw Data

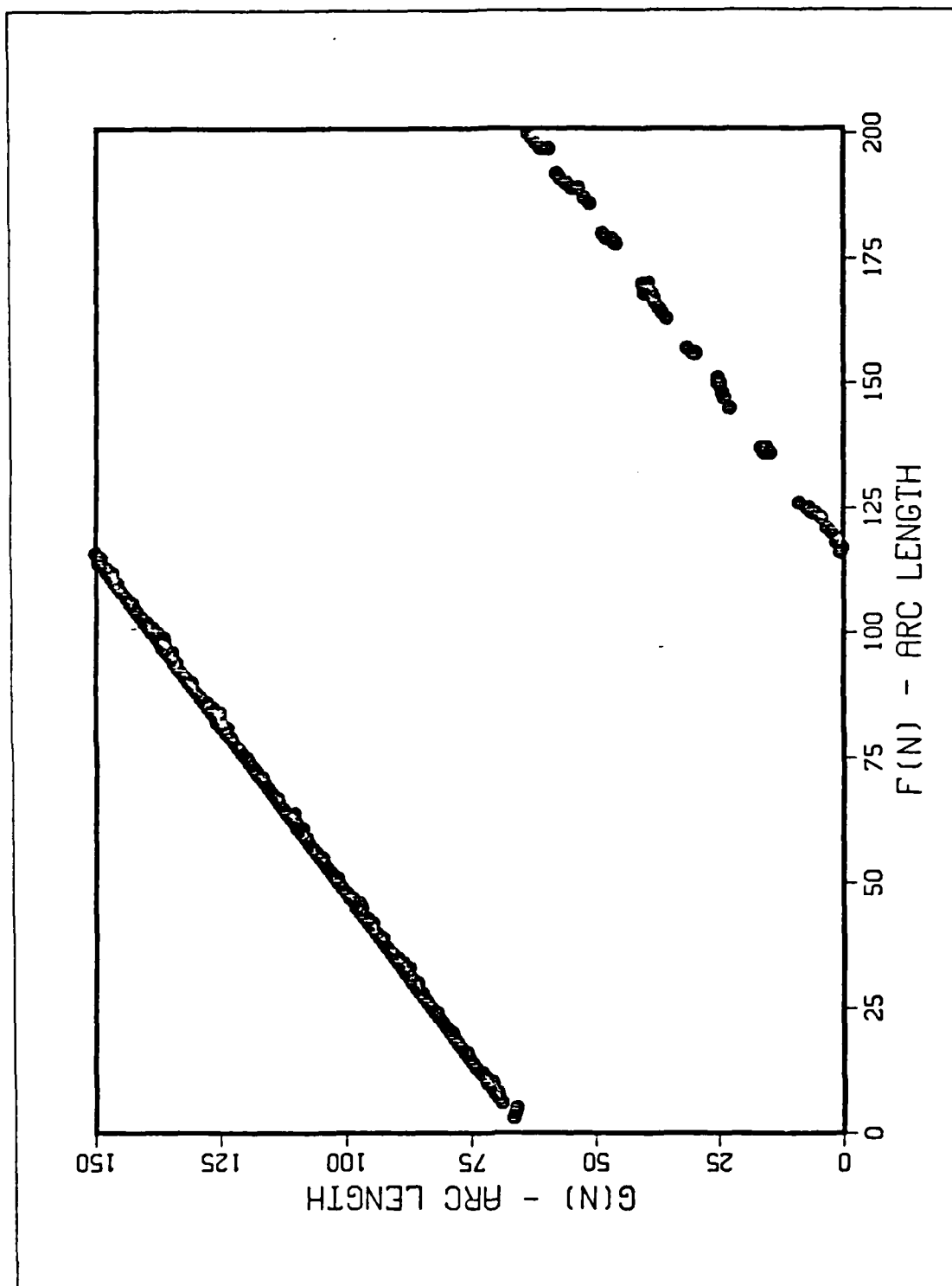


Figure 3.3 Correspondence Chart - Processed Data

to note that linear segments having negative slopes also correspond to matched sections too; if both boundaries are traversed in the same direction, these matches are not meaningful, unless one of the object happens to be 'reflected' - mirror image). This chart can be 'cleaned up' to filter out all but those points lying along the longest linear segment (with positive slope). This is shown in Figure 3.3. This figure shows that the segment from point 1 to about 120 of curve $f(n)$ matches the segment corresponding to point 60 to 150 of curve $g(n)$. It indicates that there is a poorer match over the remaining segments. It also shows that the scale difference is $120/90$, or 1.333, and that the two curves are displaced by about 60 points with respect to each other.

The above algorithm basically performs an efficient correlation over a wide range of scale. The success of the algorithm depends largely on the sophistication of the 'straight line finder' routine.

In contrast to the correlation approach, the Hough Transform matching technique is not affected by arc length variation (in the sense that arc length does not enter into its computation). This is because the Hough Transform does not make use of the ordered sequence information of the boundary points. This makes the Hough Transform sensitive to false peaks (random matches of unrelated points), but is also the reason why this technique is so much simpler. Correlation technique matches points of an ordered sequence of one curve against corresponding points of an ordered sequence of another curve. It is this need to keep the points ordered here that increases the computation burden in this technique.

D. SCALE AND ORIENTATION INVARIANT REPRESENTATION

It was obvious from the beginning that 'angle information' is scale and orientation invariant. The angle between

two straight lines remains unchanged regardless of the scale and rotation. It also became obvious, after searching for a while, that the arc length to chord length ratio between two points on the boundary (called the ACR henceforth for convenience) is also scale and orientation invariant.

This suggests the following form of representation. Code each boundary point in terms of the angle made by the tangent to this point and a specific chord. This specific chord is the chord connecting the boundary point to the nearest boundary point (in a specific direction of traversal) with the property that the ACR between these points is equaled to a pre-determined value. We shall call this the β - s representation. Figure 3.4 illustrates this. The curve is not closed to emphasis the fact that this coding scheme applies to both open and closed figures.

Implementation of the β - s representation (for ACR 1.05 and 1.3) on shapes R35-52, R34-3lp and R34-102 are given in Figures 3.5 and 3.6. Outlines of these shapes can be found in Figures 4.21 and 4.5. (For details of how these shapes are generated and the meaning behind their names, see the appendix. In Figure 3.5, the two curves have been properly scaled so that the difference in arc lengths between them are removed. This allows for easy comparison. Figure 3.6 has not been so scaled; the change in the arc length due to the noise is very evident here.

It can be seen that the representation is virtually identical over identical portion of the original shapes. The partial match between R35-52 and R34-3lp is evident. Figure 3.6 shows the effect of noise on this representation. It can be seen that small perturbation in the boundary curve can cause disproportionately large changes in the representation. This effect is localised to the neighbouring region only. Although not shown, it is obvious that this representation is independent of orientation.

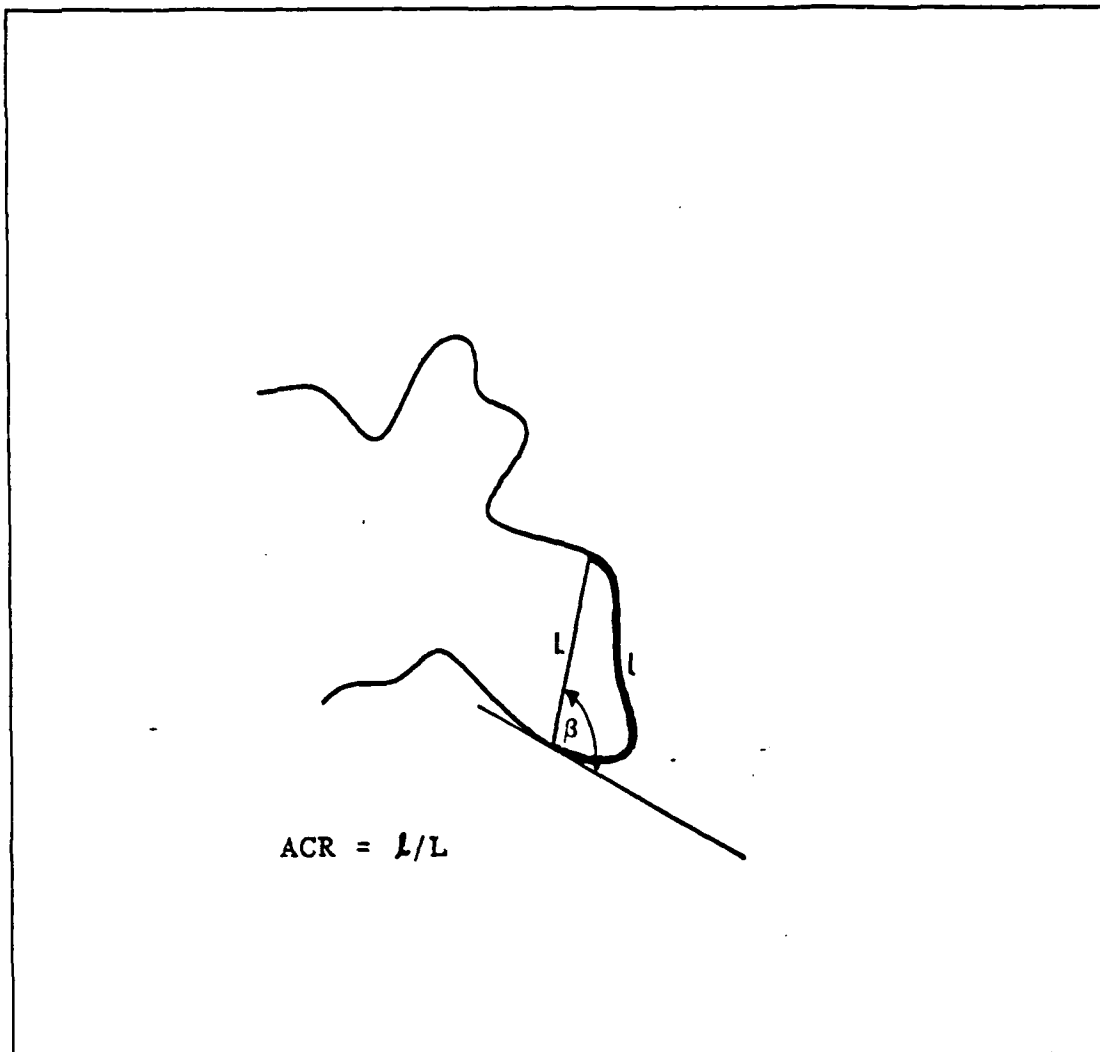


Figure 3.4 Arc to Chord Length Ratio Illustration

The ACR specification is a free parameter that can be adjusted. The larger the ACR, the larger will be the average distance between those points satisfying this ratio, ie the less 'local' the representation becomes. Also objects with relatively smooth boundaries would conceivably require a smaller ACR specification. The choice of an 'optimum' ACR may be very shape-dependent.

We note that the ACR specification is basically used to define the 'extent' of the small section of the boundary

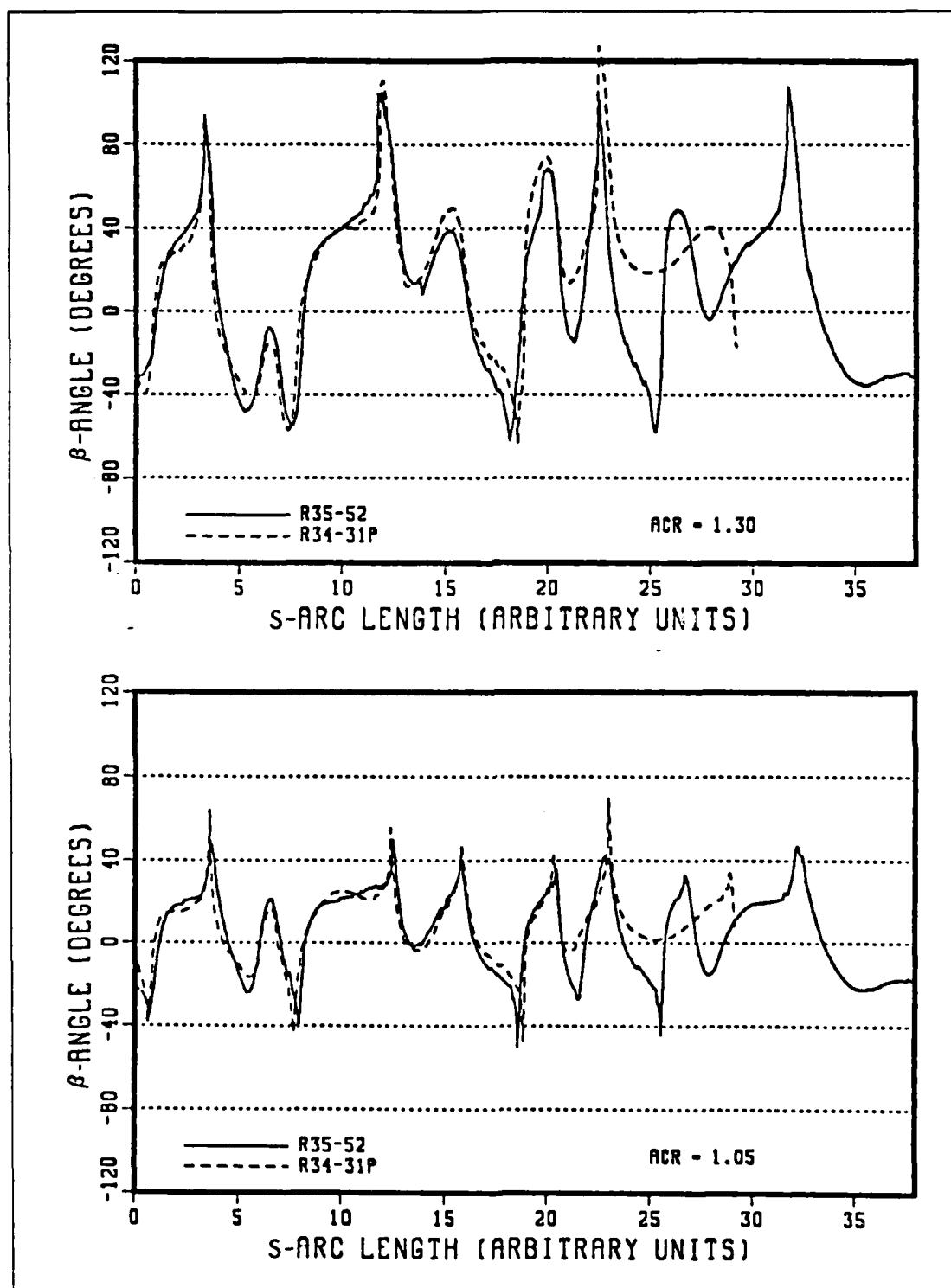


Figure 3.5 β -s Representation for R35-52 and R34-31p

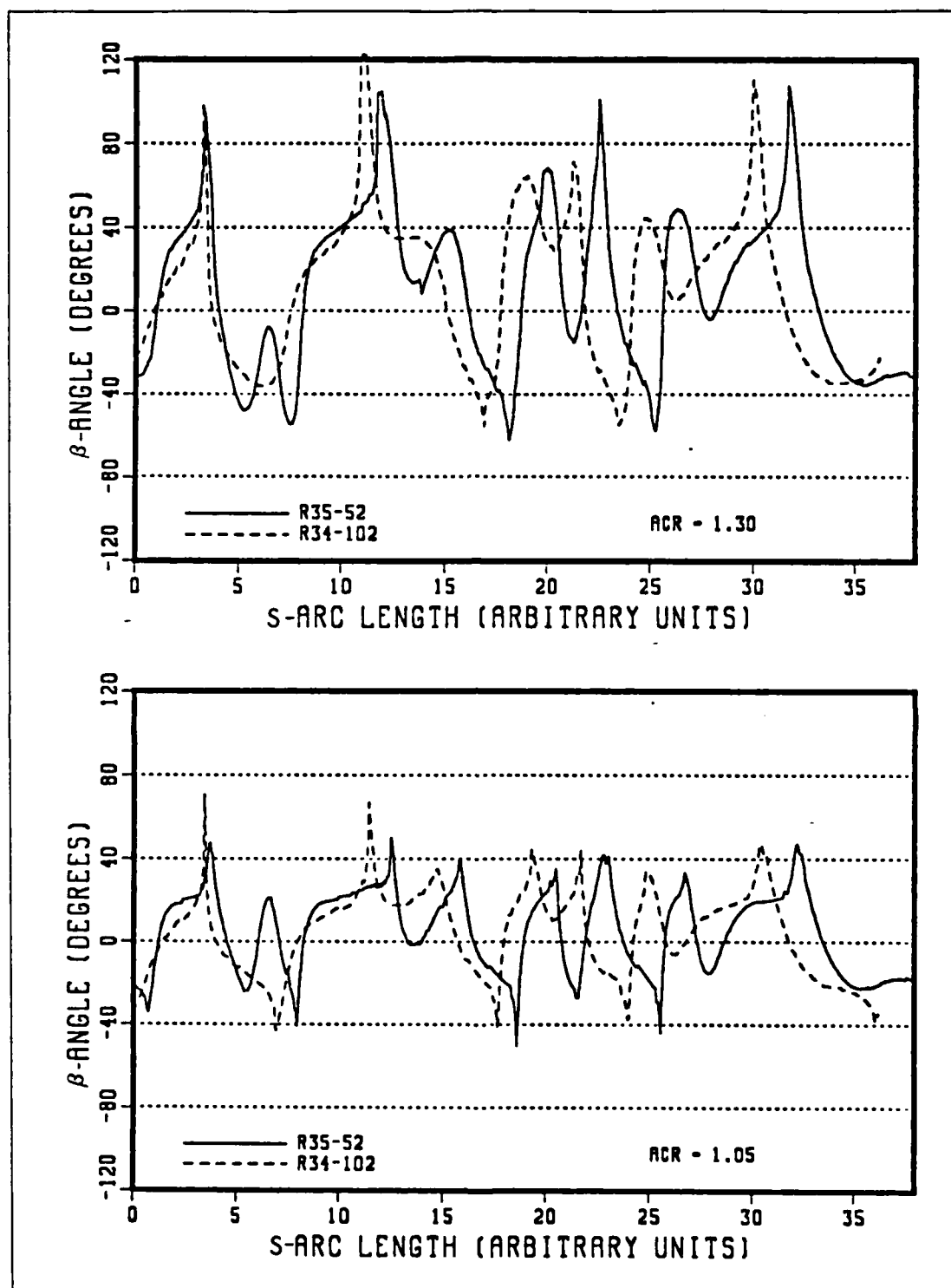


Figure 3.6 β -s Representation for R35-52 and R34-102

discussed previously. This specification is both 'local' as well as scale and orientation invariant. This is by no means the only specification available. We can develop a whole family of them. Figure 3.7 illustrates two other possible specifications. One uses the area to chord length squared ratio and the other uses the ratio between the length formed by the two tangents and the chord.

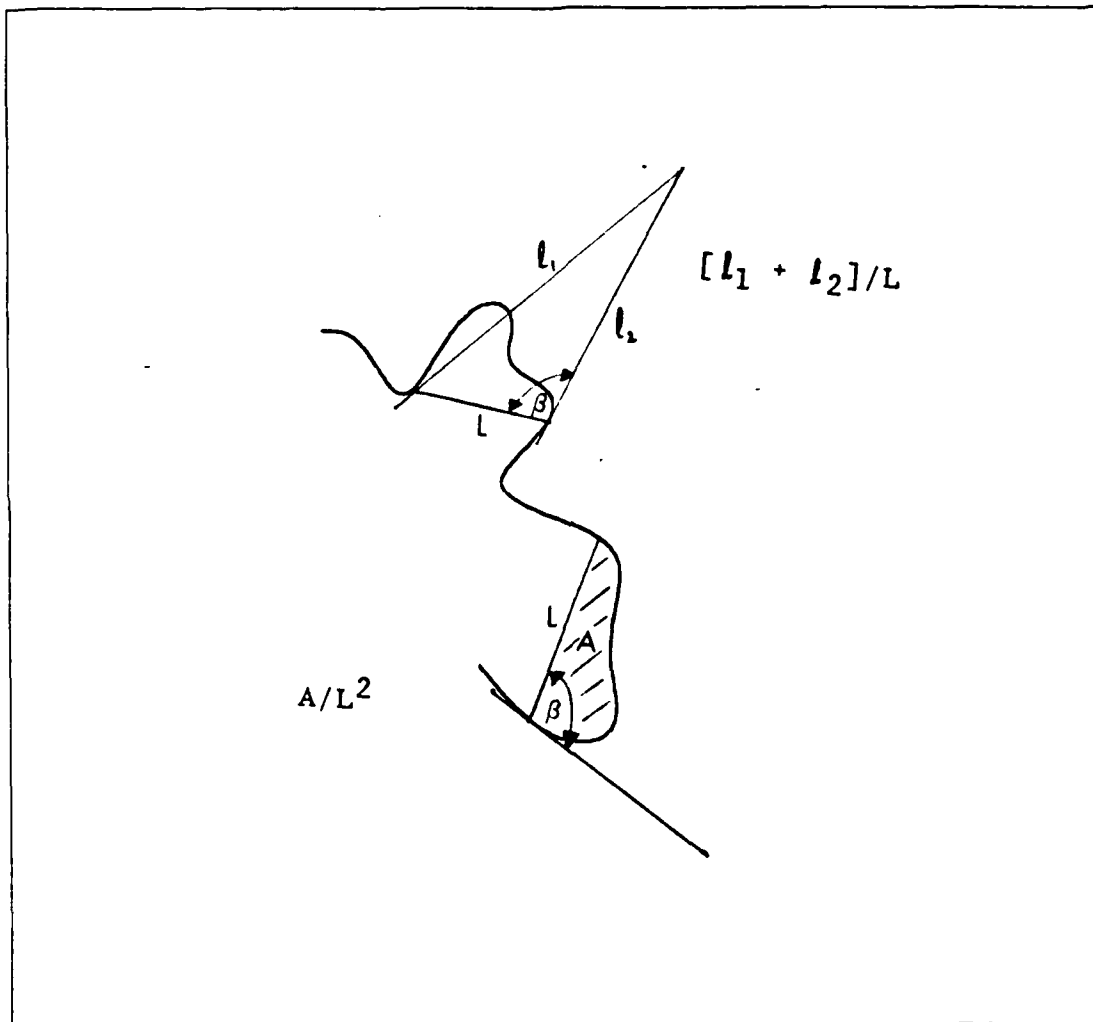


Figure 3.7 Two Other Possible Specifications Besides ACR

The sensitivity of this ACR specification is due to the unfortunate fact that geometric distortion affects the arc

length directly. Two points that originally satisfy the ACR specification in the coding phase may fail to do so in the matching phase if the segment of the boundary joining them is distorted. A small perturbation in the boundary can lead to a large change in the β coding.

E. SCALE AND ORIENTATION INVARIANT HOUGH TRANSFORM

Given the scale and orientation representation developed in the last section, we could use the 'correspondence chart' algorithm to find possible matches. However, the particular nature of this representation allows us to use the simpler technique of Hough Transform with the additional simplicity that it is scale and rotation invariant. We shall call this the $\beta - \varphi$ correlation technique. The coding and matching algorithms (using the ACR specification) are given below.

Algorithm 2: β - φ Coding

- a. Determine a reference line (usually taken to be the x-axis for convenience).
- b. For each boundary point (s_i), locate the next boundary point (s_j) (in a specific direction of traversal) such that the ACR specification is met.
- c. Determine the angle (β) between the chord joining s_i to s_j and the tangent to s_i . The sign of this angle is positive if the segment of the shape bounded by this points is convex, and negative otherwise.
- d. Determine the angle (φ) between the chord and the reference line, measured clockwise from the reference line (see Figure 3.8).
- e. Determine other independent relation(s) between s_i and s_j . For instance, the angle (α) between the tangent lines to these boundary points.
- f. Code each boundary point in terms of the vector r , where $r = (\varphi, \alpha)$. Set up a R-Table relating β to (φ, α) . The Table is indexed by β (Table 2).

Algorithm 3: β - φ Matching

- a. Set up an accumulator $A(i)$ of N elements, where N is the number of the (discretized) possible orientations of the reference line. (Thus $N = 36$, if each orientation is 10 degrees wide). Initialize accumulator to zeros.
- b. For each boundary point on the test shape, obtain $\hat{\beta}$, and $(\hat{\varphi}, \hat{\alpha})$.
- c. For each pair of $(\hat{\varphi}, \hat{\alpha})$ indexed by $\hat{\beta}$ in the R Table, check if the independent relation matches. If it does, then determine the possible orientations (θ) of the reference line from $\hat{\varphi}$ and φ . Increment the corresponding element of the accumulator. If not, proceed on to the next boundary point. In other words,

```
if  $|\hat{\alpha} - \alpha| < \text{tolerance}$ 
  then
     $\theta = \hat{\varphi} - \varphi$ 
     $A(\theta) = A(\theta) + 1$ 
  else
    next boundary point
```

The peaks in the accumulator array then correspond to possible matches of the two shapes. The locations of the peak in the array indicates the most likely orientations of the reference line, and thus correspond to the relative orientations between the two shapes.

For ease of future reference, we shall call the specification used to pair the two points (s_i, s_j) as the primary specification, and the additional specifications used to relate these points as the secondary specification(s). Also the pair of points (s_i, s_j) shall be called the coded pair. We shall use β to represent the coded information based on the primary specification, α to represent the further constraints based on the secondary specification(s) and φ

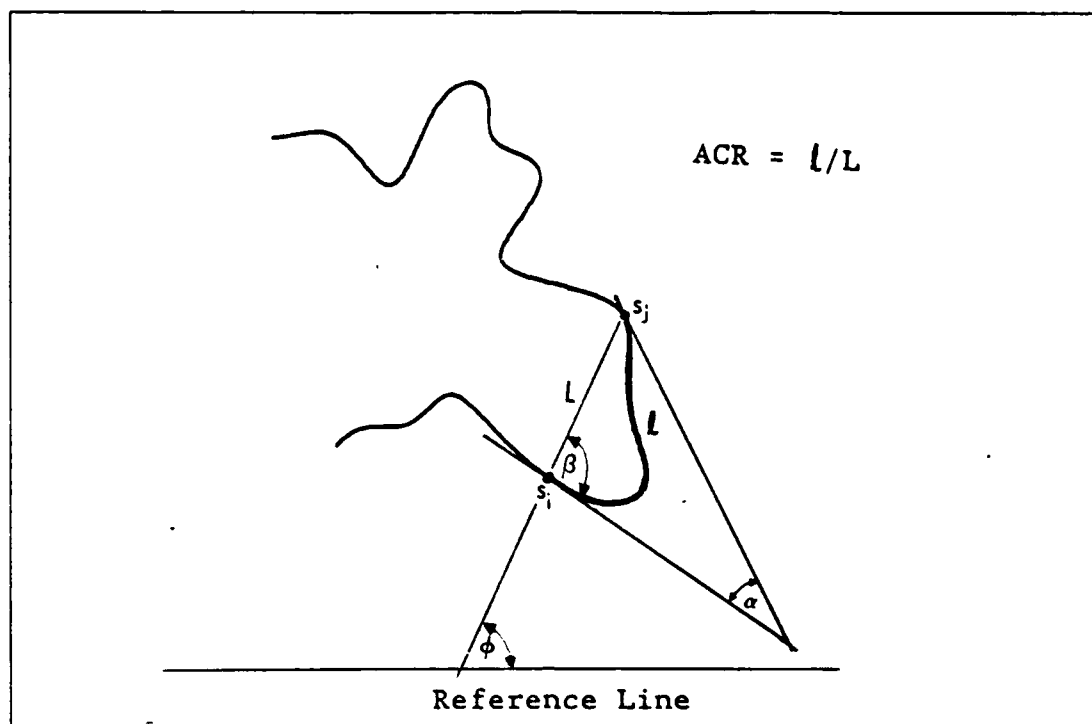


Figure 3.8 β - ϕ Coding

to represent the angle between the reference line and the chord joining the points in the coded pair.

This technique differs from the basic Hough Transform in two essential ways. Firstly, this uses a reference line whose orientation is to be reconstructed, rather than a reference center whose coordinates have to be determined. Secondly, each boundary point is identified by β , which is local (referenced to the local tangent) rather than the gradient angle, which requires an external reference axis. These two differences make this matching technique scale and orientation invariant. Another distinction is the use of an independent relation (α). By only using those points that are simultaneously related in both the β and α parameters, we reduce a fair portion of accidental matches. Of course we could use more independent relations to further restrict the possible match points. The limitation will be the

TABLE 2
R-TABLE FOR $\beta - \phi$ CODING

Angle between Chord and Tangent to Boundary Point	Set of Vectors $r = (\phi, \alpha)$
β_1	$r_{11}, r_{12}, \dots, r_{1n}$
β_2	$r_{21}, r_{22}, \dots, r_{2p}$
.	.
.	.
.	.
β_m	$r_{m1}, r_{m2}, \dots, r_{mq}$

number of possible independent relations available (which must be scale and orientation invariant and relatively insensitive to noise). The tolerances set on these specifications will determine the sensitivity to geometric distortion. The smaller the tolerance, the more sensitive it becomes. The tolerance must obviously be tighter for the primary specification than for the secondary specifications.

This scheme is applied to shapes R35-52, R34-3lp and R34-102. The results, using 2 different values of ACR are shown in Figures 3.9, and 3.10. The accumulator values are normalised by dividing the values by the number of points on the test curve. (In all the examples in this report, the test curve is that given by the dashed line). These values can be easily interpreted as correlation coefficients. For example, Figure 3.10 indicates that at zero relative orientation of the 2 shapes, about 40% of the points in the test shape can be correlated with points in the reference shape.

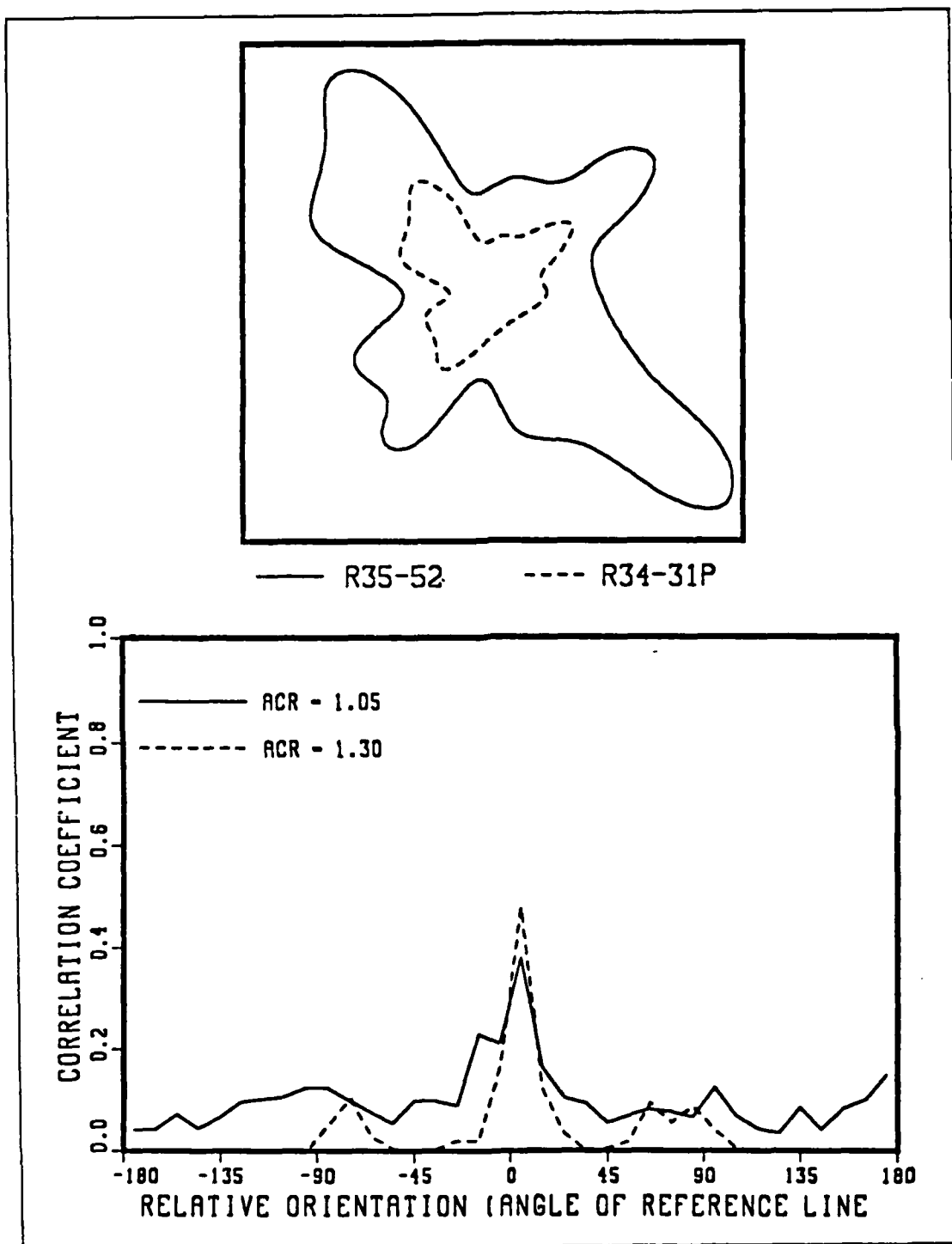


Figure 3.9 Matching of R35-52 and R34-31p Using β - ϕ Correlation with ACR Specification

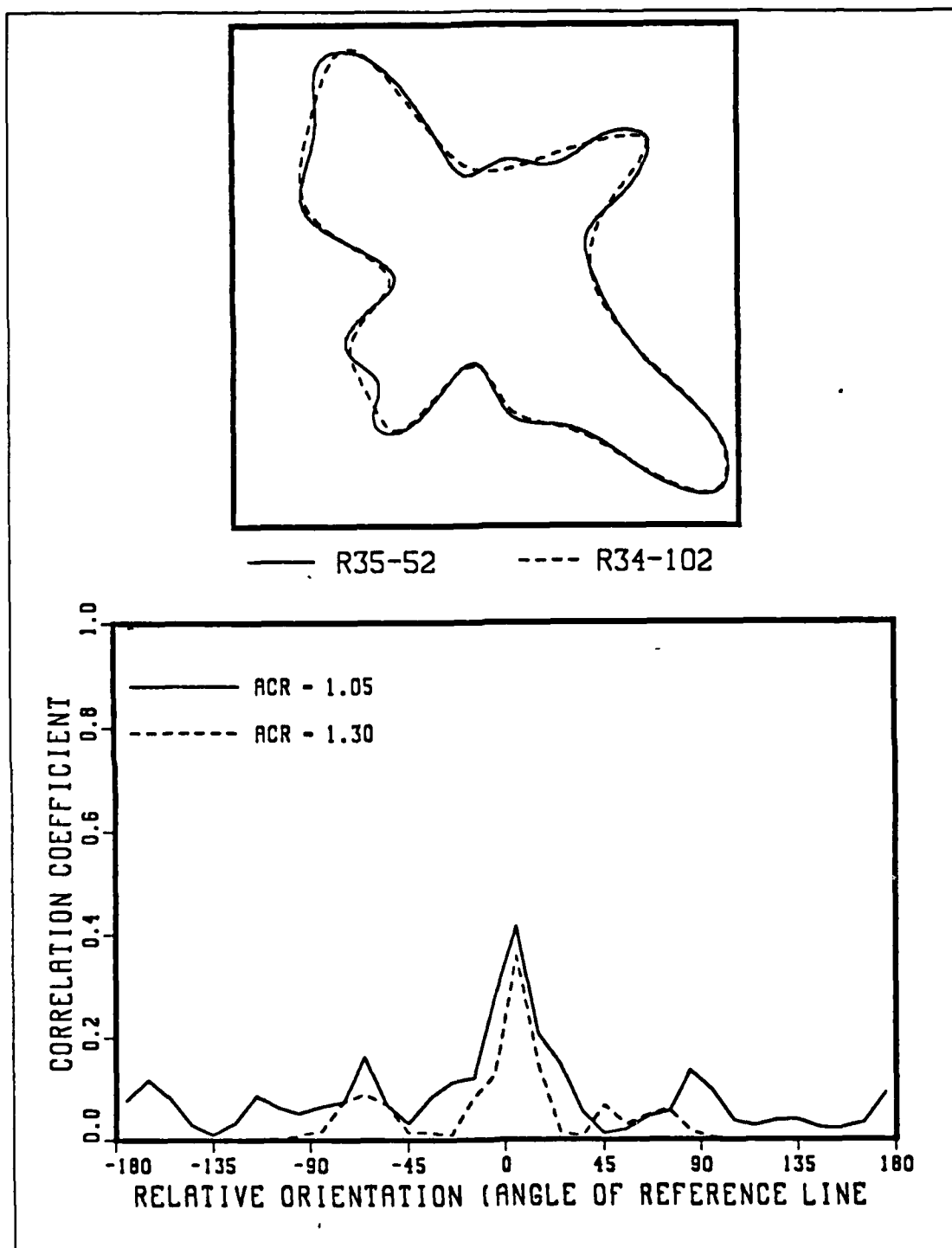


Figure 3.10 Matching of R35-52 and R34-102 Using β - ϕ Correlation with ACR Specification

By the nature of the coding this correlation is not point to point correlation, but rather point-on-a-segment to point-on-a-segment correlation; ie, the correlation is made on the basis of the behavior of the boundary in the vicinity of the point. Visually, we can see that the correlation should be higher than this. The low correlation is a direct consequence of the sensitivity of the ACR to geometric distortion. Both figures, however, correctly indicate that the best correlation between the shapes being tested occurs at zero degree relative orientation.

To improve the correlation, we need to make the β - ϕ coding less sensitive to noise. This implies that we need alternative primary specification and, perhaps, secondary specifications too. The other possible specifications mentioned earlier were tried and found to be unsuitable too.

In the next chapter, we shall describe a new primary specification that is less sensitive to the effects of noise. Using this, the resulting correlation between R35-52 and R34-102 increases to 80% (see Figure 4.5). To do this we need to forgo the demand for scale and orientation invariance. However, the matching algorithm can be easily modified to enable the algorithm to match shapes of arbitrary scale and orientation with a slight increase in computation.

IV. A NEW CORRELATION TECHNIQUE

A. INTRODUCTION

The algorithm developed in the previous chapter is sensitive to noise. This is due to one main reason. We have removed the scale unknown by using the ACR measure; arc length is, unfortunately, very sensitive to geometric distortion. In other words, we have replaced an unknown factor with an uncertain measure. Thus, unless, we can find an alternative measure that is scale independent and reasonably immune to noise, this approach may be of limited practical use. Such a measure was not found.

We therefore remove the scale invariant constraint. What we eventually found is a new and interesting approach to boundary coding. In its essence, each boundary point is coded with respect to another point picked at random from the boundary. Note that this coding is not scaled and orientation invariant. In fact identical shapes would yield different codes if different sets of random numbers are used!

B. RANDOM CODING

We used as primary specification, the random separation between the coded pairs. The property coded at each point is again β , the angle between the tangent to this point and the chord joining the coded pair. To retain the 'local' features (essential for partial match applications), the range of the allowable separation (called the coded range henceforth) is restricted. For illustrative purposes, 3 sets of coding ranges are used in the examples below, namely 10 to 60 points, 80 to 130 points and 150 to 200 points (i.e., the second element in the coded pair is picked from any point that lies between 10 to 60 points away from the first element, etc).

Two secondary specifications are used: the ACR and the angle made by the tangents to each point in the coded pair (ie. α in the previous algorithm). In the matching process, since the points are paired randomly, it becomes necessary to check each point against all other points in the test shape. In practice, since the coding range is itself restricted, this process can be also restricted to a smaller section of the boundary. In the examples that follows, this search range is limited to half the entire boundary length. Further savings in computation is achieved by checking only alternate points within this range.

The basic algorithm for this technique is similar to the previous one. For clarity, we shall restate it. Note that β and ϕ below refer to the same angles as in the previous algorithms, while α is used differently here.

Algorithm 4: Improved β - ϕ Correlation

- a. For each boundary point in the reference shape, select another boundary point at random from those within the allowable range. Determine the (β, ϕ, α) relation between the coded pairs thus found. (Note: α contains two components, the ACR and the tangent angle measures). See Figure 4.1.
- b. Construct the R-Table in the same manner as before.
- c. Initialize the accumulator array as before.
- d. To match a test shape, determine for each boundary point, the $(\hat{\beta}, \hat{\phi}, \hat{\alpha})$ relation with all other boundary points within the search range. For each $(\hat{\beta}, \hat{\phi}, \hat{\alpha})$ and corresponding (β, ϕ, α) from the R-Table, reconstructs the reference line as before.
- e. Peaks in the accumulator array correspond to possible matches of the two shapes with the location of the peaks corresponding to the relative orientation of the two shapes.

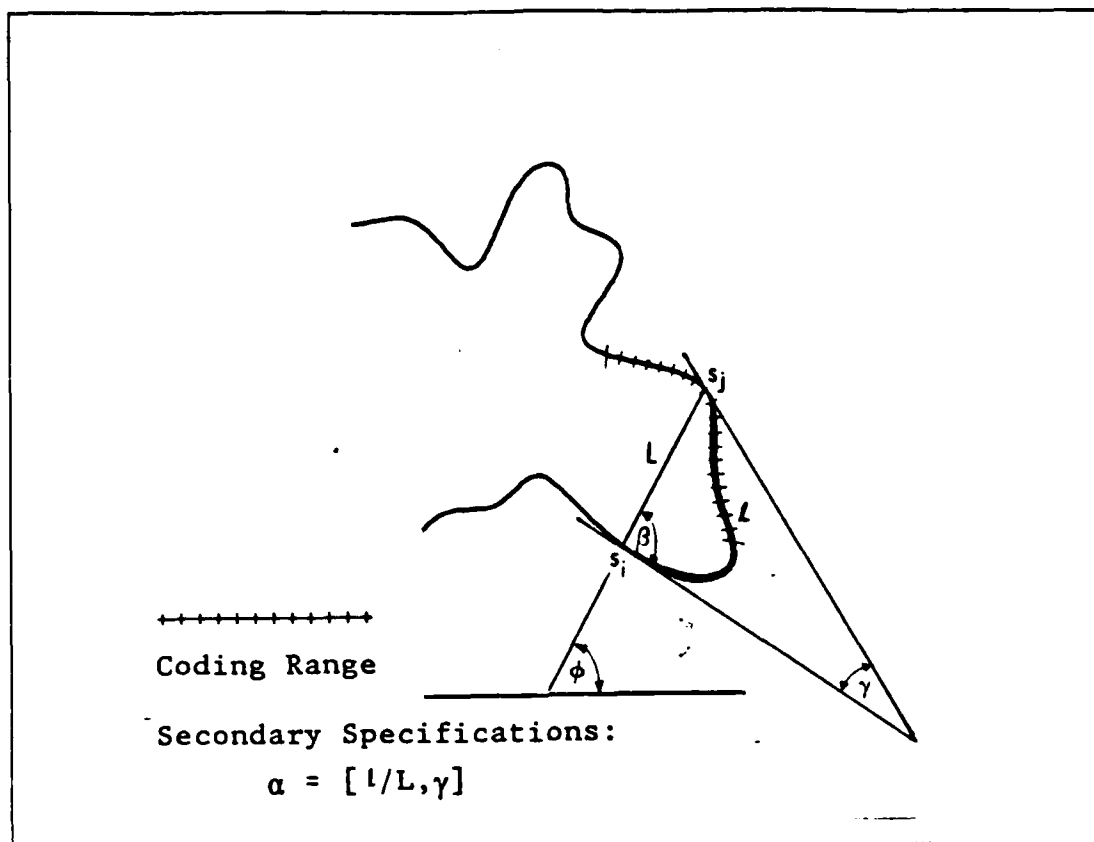


Figure 4.1 β - ϕ Coding Using Random Separation

We shall discuss the key features of this technique and provide heuristic explanations, where possible, on the 'hows' and 'whys' of it. These features are verified in the numerous examples that follows.

C. FEATURES

1. Scale and Orientation Invariance

The coding is not scale and orientation invariant. The scale unknown is resolved in the matching algorithm by pairing each point with all other points within the search range. This, in essence, performs a matching over a range of scale. The orientation unknown is not a problem, since the output of the matching process will indicate the relative orientation of the two shapes. The correlation is

performed, in essence, over the range of possible relative orientations. In this respect, this correlation technique is not affected by unknown scale and orientation and can be said to be invariant to these.

2. Robustness

The random separation helps to 'break' down the effects of noise. Consider the alternative of using a fixed separation, say n . Then if the coded pair (s_i, s_{i+n}) is affected by noise, the next pair (s_{i+1}, s_{i+1+n}) is likely to be similarly affected. However, if the separation is random, and if (s_i, s_j) is affected by noise, it is not necessary that (s_{i+1}, s_k) (where j and k are randomly picked) would also be affected. More importantly, even if it is, the effects in the two coded pairs are unlikely to be the same, ie. the false matches they cause are not likely to be correlated.

For the case of fixed separation, because of the strong correlation (close proximity) between the coded pairs, noise in their coding are likely to be correlated, giving rise to false 'peaks' during the process. This implies that in order to achieve the best decorrelation of false matches, the boundary should be coded such that the parameter, β , is uniformly distributed across its range, -180 to $+180$ degrees. This may require extending the coding range to a substantial fraction of the entire boundary length, which may not be always desirable since the coding then becomes less 'local' in nature.

Another factor that helps to reduce the effects of noise is the nature of the matching algorithm. Figure 4.2 illustrates this. The solid line there refers to portion of the reference shape and the dashed line to the test shape. Point s_i is paired with s_j during the original coding. In the matching algorithm, since s_i is paired with all other points, it would be eventually paired with one that is close

geometrically to the original s_j (ie. \hat{s}_j in Figure 4.2) and that also satisfy the secondary specifications. Thus, we would expect to recover the orientation of the reference line.

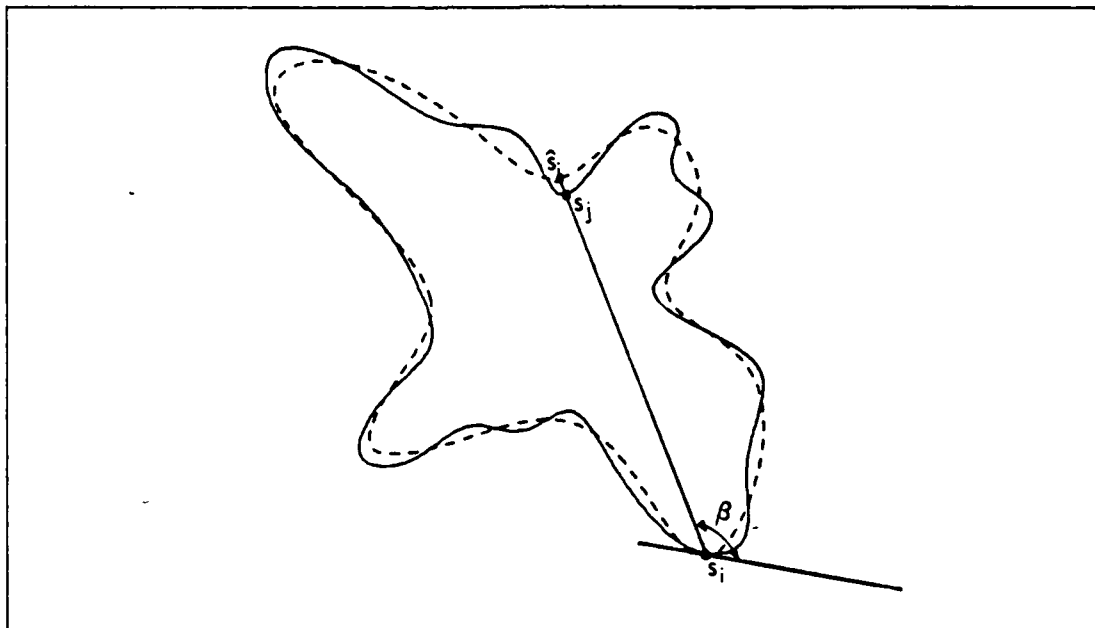


Figure 4.2 Matching in the Presence of Noise

3. "Local" Characteristics

The choice of the coding range determines the amount of 'local' information captured in the coding. The lower the upper limit of the coding range, the more 'local' the representation becomes. If the coding range is the entire boundary, then the coding takes on a global nature. This will be clearly illustrated in the examples on Partial Matching below.

4. Discrimination

The distance of the coding range from the point being coded also determines the level of discrimination in the matching process. The closer this distance is, the smaller the segment the matching algorithm would be trying

to find matches. What is important here is the fact that small segments tend to look more similar than larger segments. Thus, a small segment from any curve would tend to look like a linear segment. Discrimination of two shapes cannot be reliably done at too small a scale. This also implies that the lower limit of the coding range should be as large as the longest linear segment of the shape, if the matching process is not to be overwhelmed by matches of short linear segments.

The algorithm uses the secondary specifications to reject obvious false matches. The types of discrimination possible with our choice of specifications is illustrated in Figure 4.3. If scale information is also available, then it can be effectively incorporated as an additional specification. An important observation is that the tolerances set on these specifications determine the 'noise rejection threshold'. The larger the tolerance, the better the matching (detection probability) under noise; the higher too would be the amount of false matches (false alarms). The tolerances used in most of the examples below are 0.1 for the ACR measure and 5 degrees for the tangent angle measure.

The reader may wonder why do we use the ACR specification when it has been stated that this specification is too sensitive to geometric distortions. There is a distinction between the role ACR play in the previous algorithm compared to the present. Previously it was used as a primary specification, whereas here it is used only as a confirmatory specification; the tolerance on it is therefore looser here, making it less sensitive to noise.

5. "End Losses"

The 'look forward' characteristics in the coding process means that the output matched segments tend to be shorter than the actual match in the input segments. This is because the section 'forward' of the points being matched

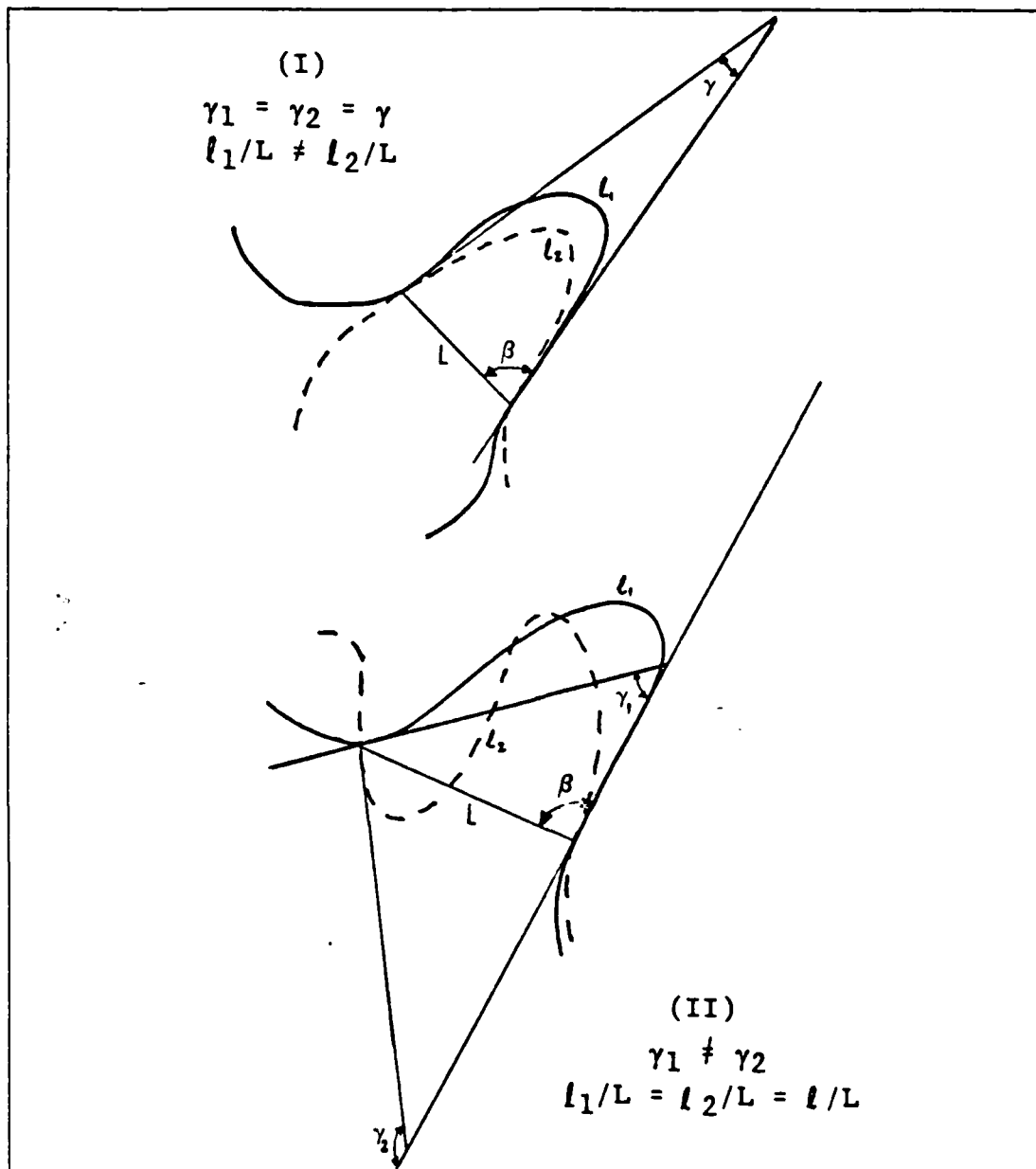


Figure 4.3 Discrimination Using Secondary Specifications

must itself matches before the 'current' segment can match. This will be clearly illustrated in the examples on Partial Matching too. The loss of the 'forward ends' can be easily removed if the coding and matching are performed in both directions.

6. Lack of Internal Consistency Checks

When matched segments of the shapes are found, the present algorithm simply counts the number of points in these segments and expresses this as a fraction of the number of points in the test boundary. It does not check to see if the relative positions of these segments in the test and reference shapes are consistent. This additional check should eliminate false matches too. This is the main weakness of this technique. Such a check could be implemented (similar to those used in hierarchical search). It has not been done to keep this basic algorithm simple.

D. RESULTS

The algorithm is applied to numerous test shapes below. These examples verify the various comments made above. It is hoped that the large number of test cases would give the reader confidence in the use of this new technique. In the examples, the number of points in the shapes are varied to ensure that any scale information that may be implicitly present are removed. As a reminder, the second number in the shape title indicates the number of points in that shape. Thus R35-52 has 500 points. Appendix A contains more details of these shapes.

In the discussion and figures that follow, N refers to the number of sample points in the test shape, and RTOL and GTOL refer to the tolerances set in the ACR and tangent angle specifications respectively. One final note before we see the results. The direction of the orientation angle is as follows. A positive relative orientation of, say 90 degrees means that the test shape (dash line) is rotated 90 degrees counterclockwise from the reference shape.

1. Geometric Distortion

To study the sensitivity of this technique to noise, we introduce distortion at varying levels into the test shapes. Figures 4.4 to 4.7 show the results for one set of

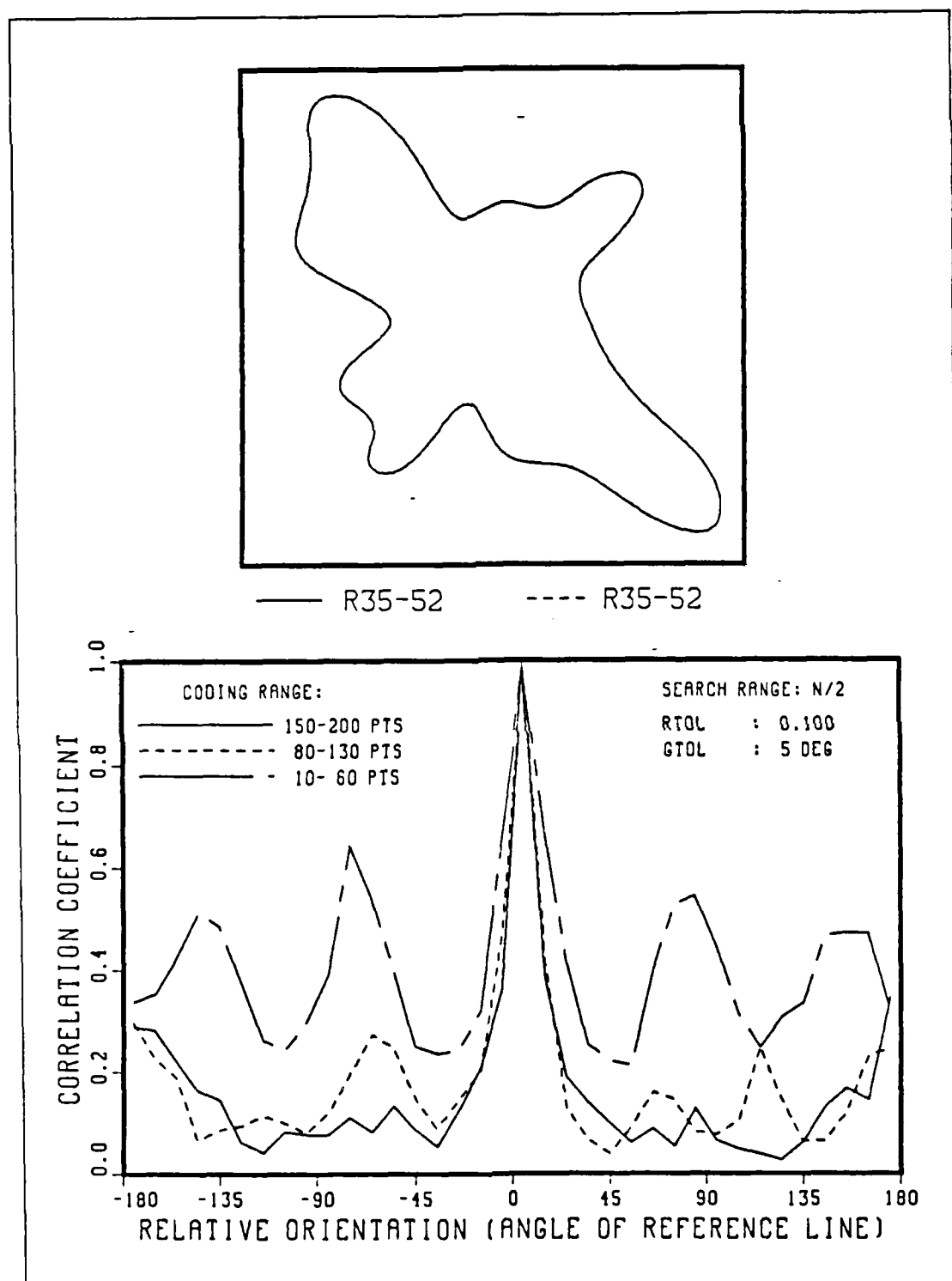


Figure 4.4 Correlation Between R35-52 and R35-52

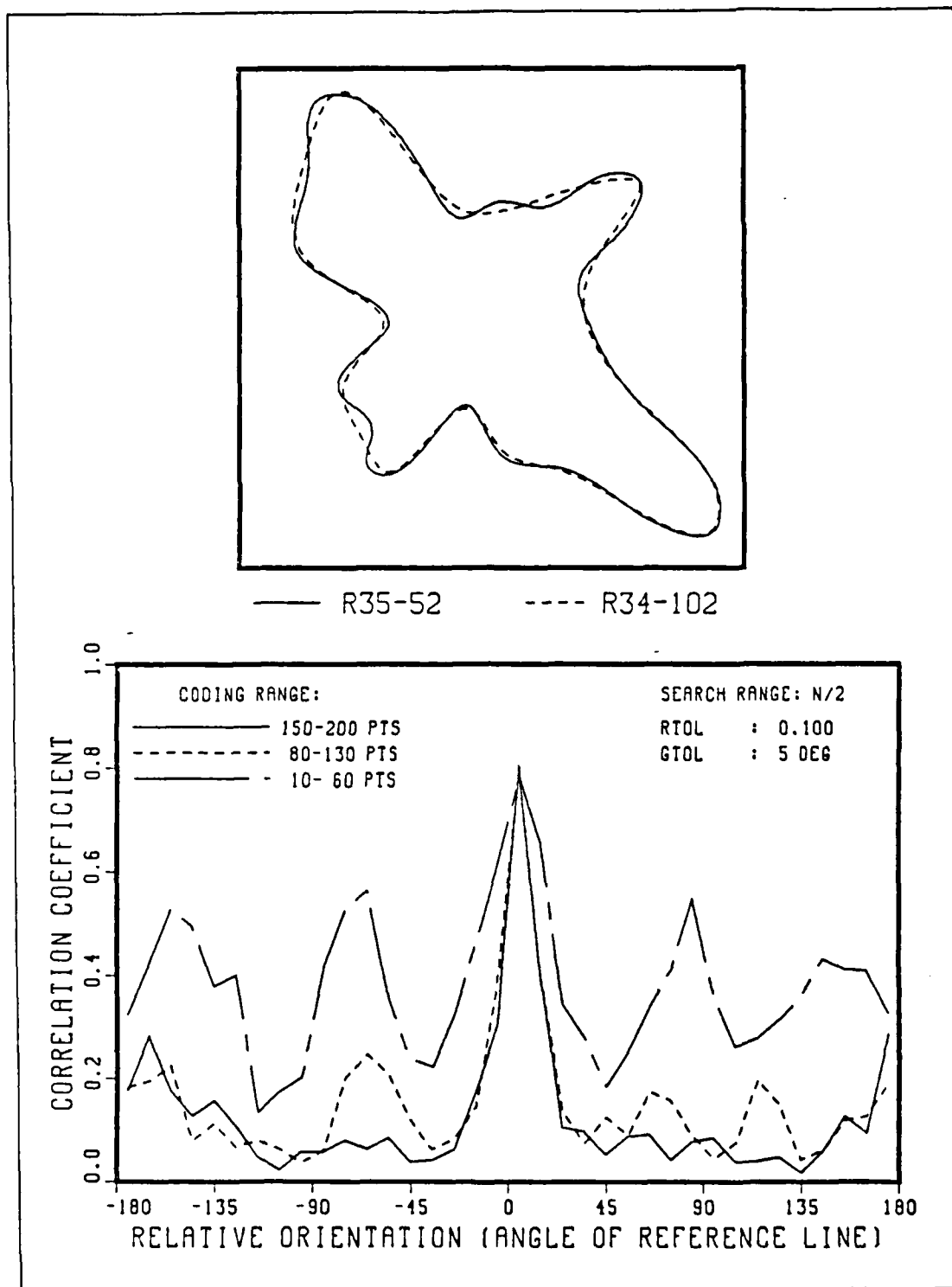


Figure 4.5 Correlation Between R35-52 and R34-102

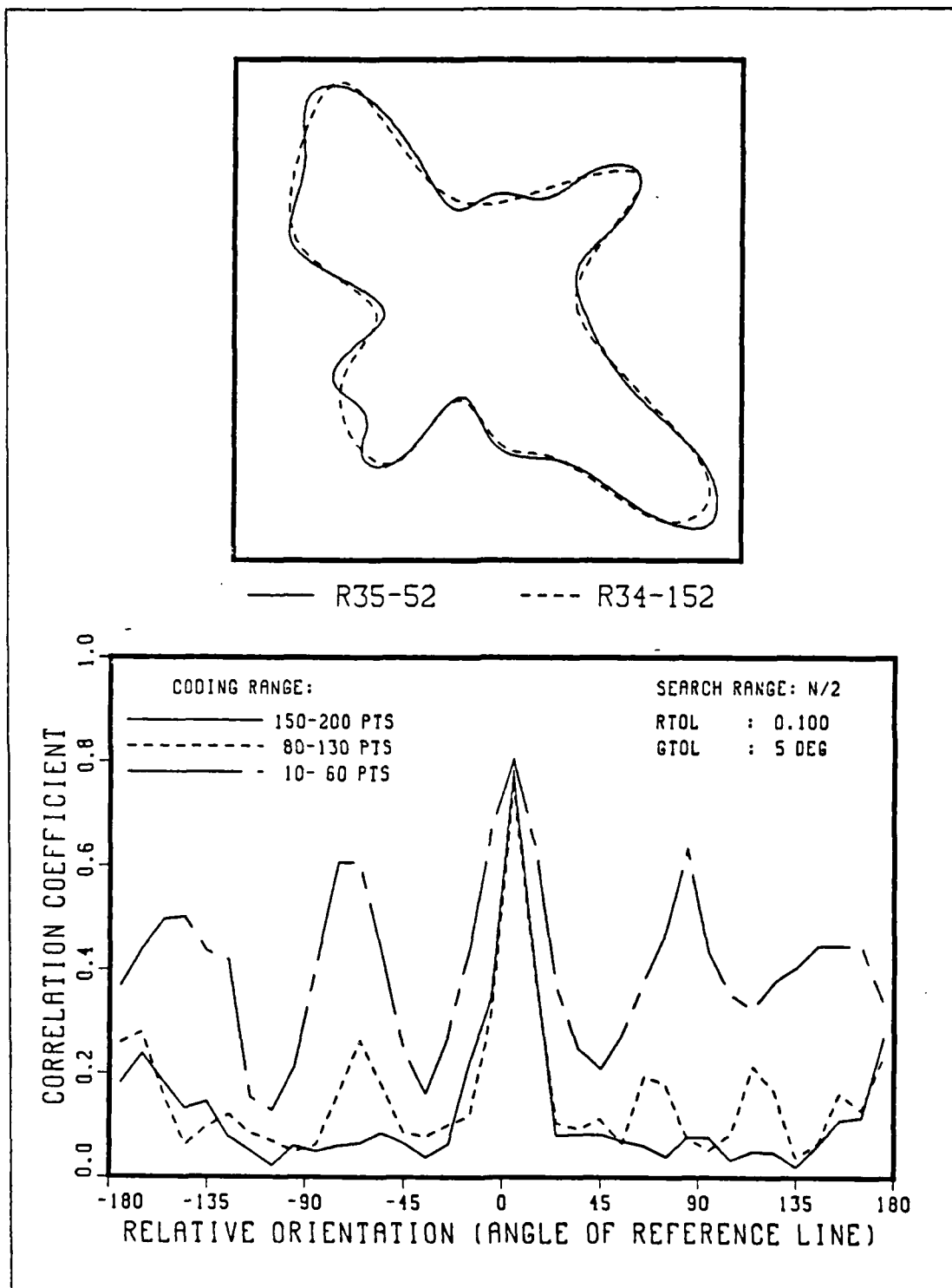


Figure 4.6 Correlation Between R35-52 and R34-152

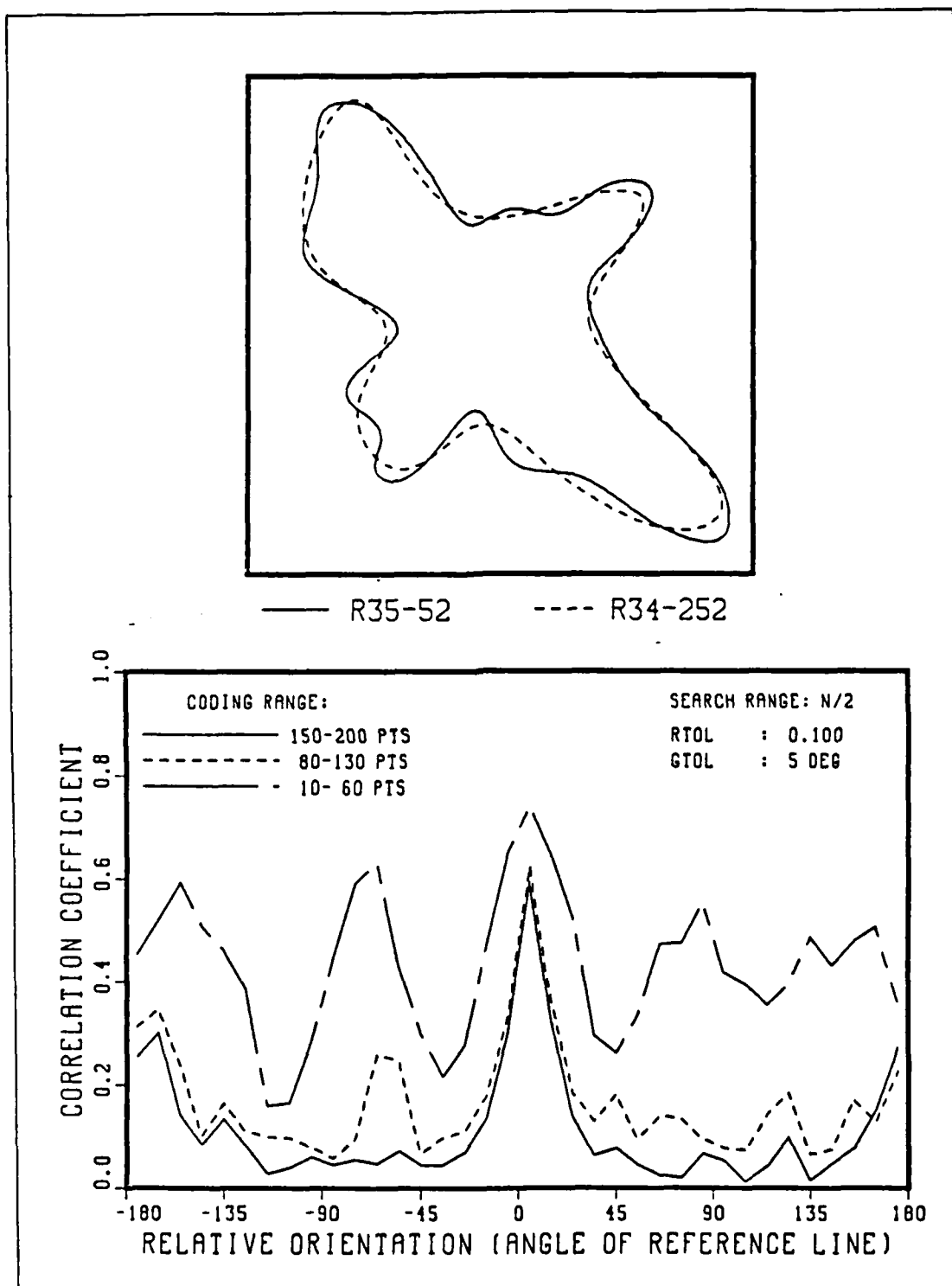


Figure 4.7 Correlation Between R35-52 and R34-252

shapes. When the two shapes are identical, correlation is 100% as expected (Figure 4.4). As the amount of distortion increases, the level of correlation decreases, until it reaches 60% for Figure 4.7. However the correlation level away from the peak value remains relatively constant, illustrating the fact that matches at these orientations are random in nature. Note also that the lower coding range (10 to 60 points) produces more apparent matches, since smaller segments tends to match better than larger segments. The correlation peak occurs at the correct relative orientation, ie. zero degree, since the two shapes are identically oriented. The result for Figure 4.5 should be compared against Figure 3.10 which uses ACR as the primary specification. This produces only 40% correlation between the two shapes. Using random coding, the correlation has increased to 80%.

The next figure, Figure 4.8, is almost identical to Figure 4.4 despite the fact that the search range has been increased from $N/2$ to $N-1$. The fact that searching through a larger search range does not produce significantly more correlation attests to the 'noise' rejection capability of the algorithm.

The algorithm is next applied to a set of more 'difficult' test shapes. Figures 4.9 to 4.12 show the correlation when the test shape is scaled down, rotated and distorted. In spite of the scale and orientation differences, the algorithm correctly locates the match at 90 degrees relative orientation. More significantly, the amount of correlation is not unreasonable compared to what one might estimate visually. For Figure 4.12, the distortion has more or less made the test shape symmetrical. It is thus not surprising for the algorithm to locate two peaks at plus and minus 90 degrees.

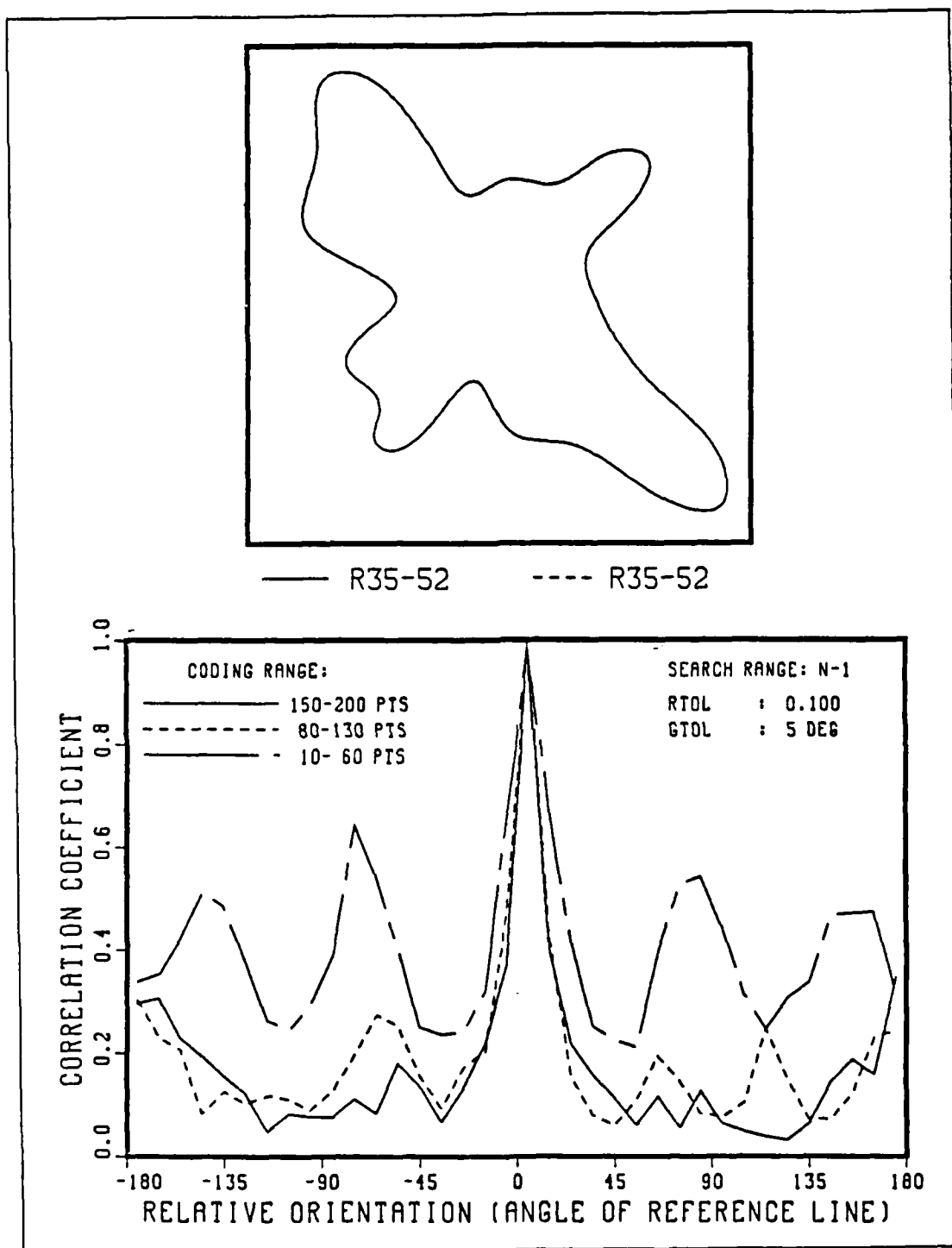


Figure 4.8 Correlation Between R35-52 and R35-52
Using a Wider Search Range

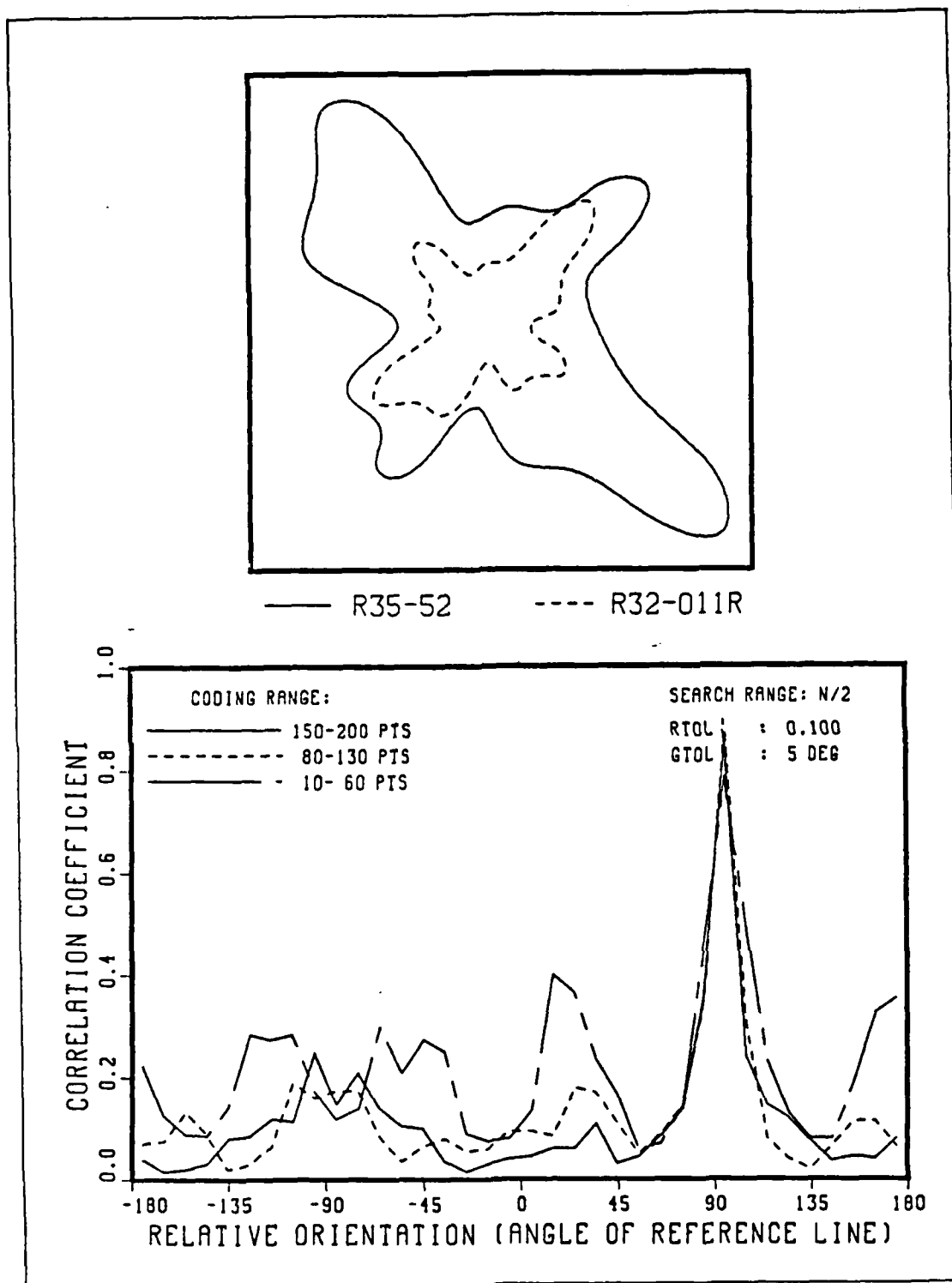


Figure 4.9 Correlation Between R35-52 and R32-011r

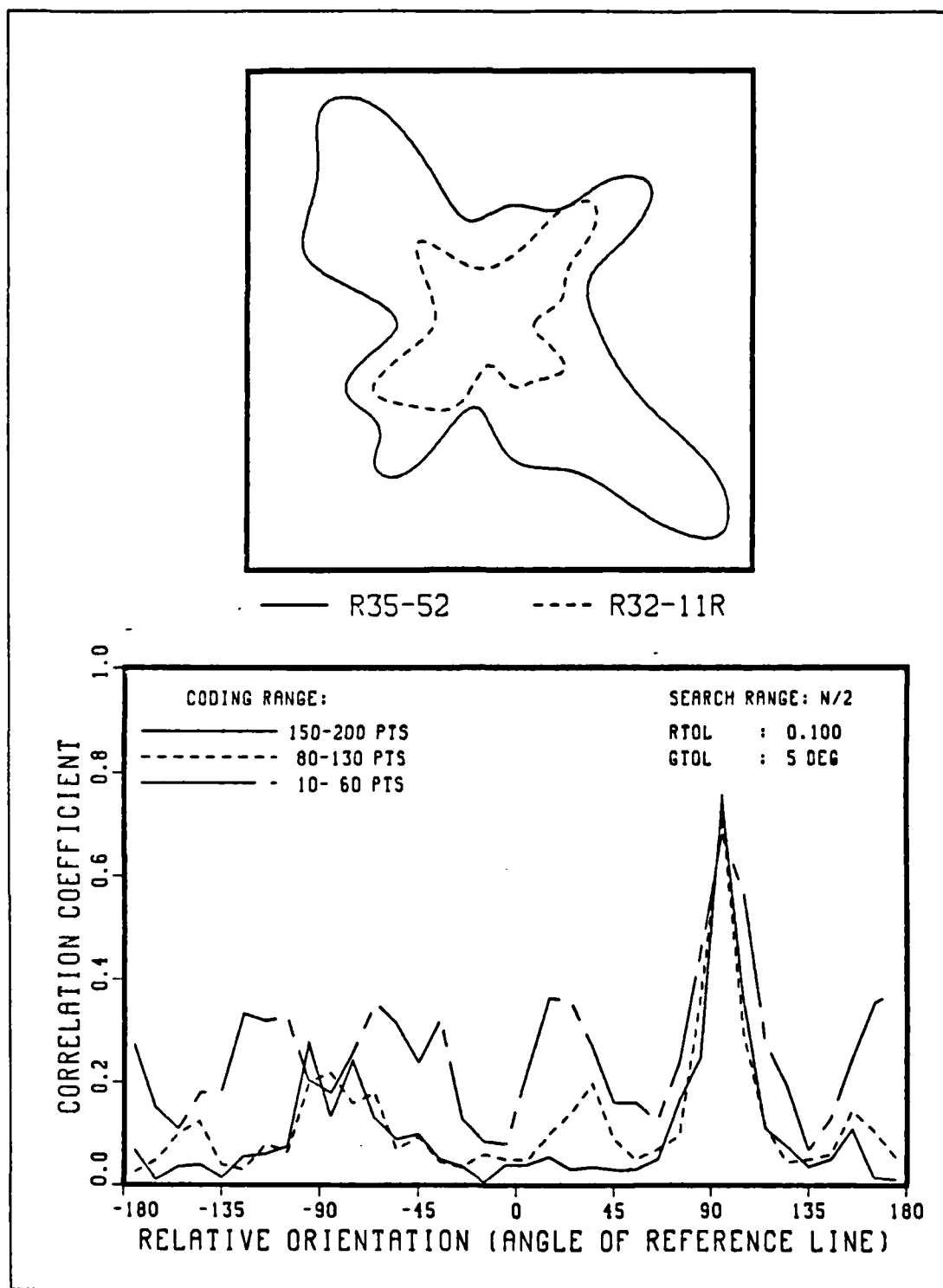


Figure 4.10 Correlation Between R35-52 and R32-11r

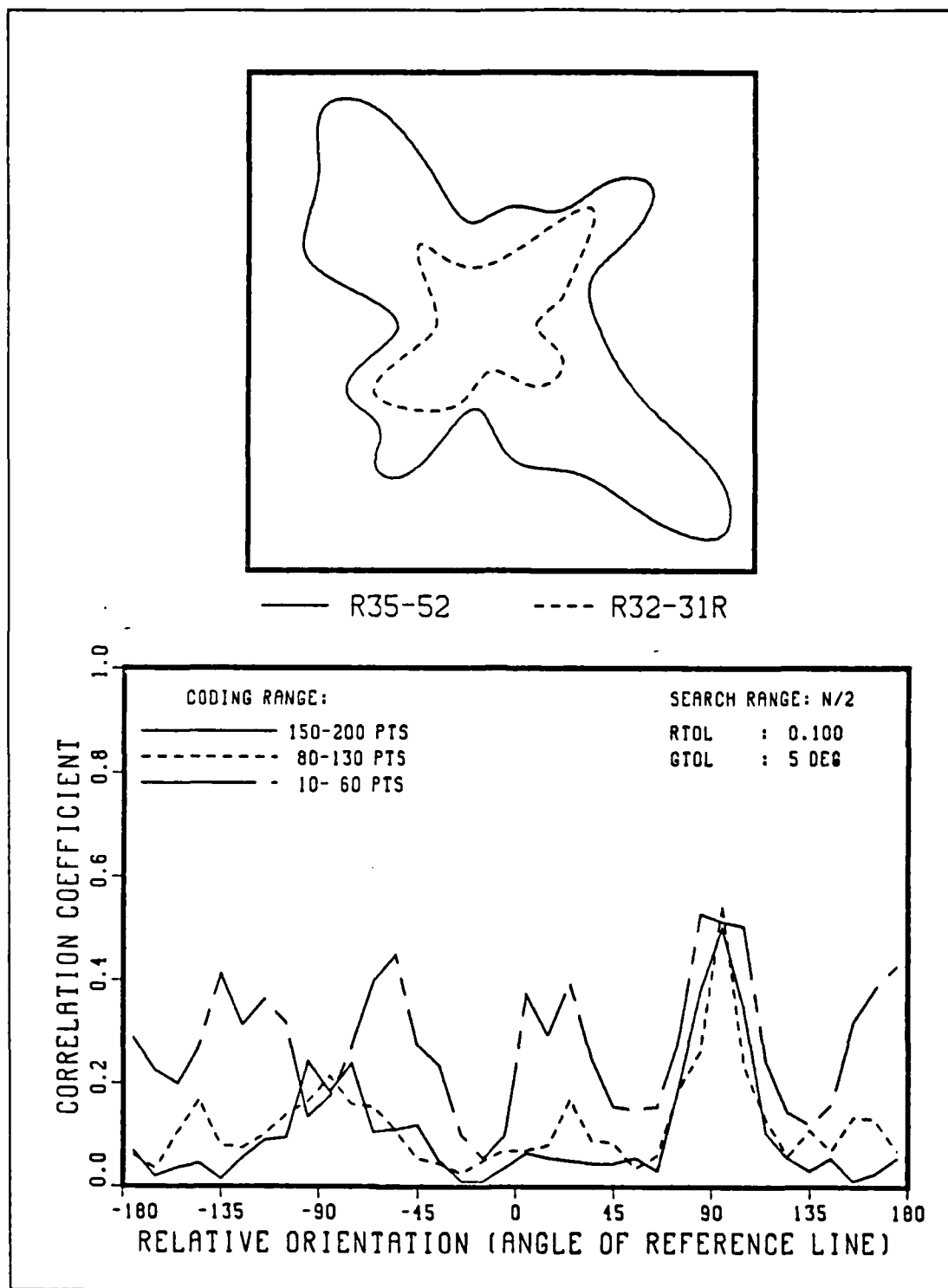


Figure 4.11 Correlation Between R35-52 and R32-31r

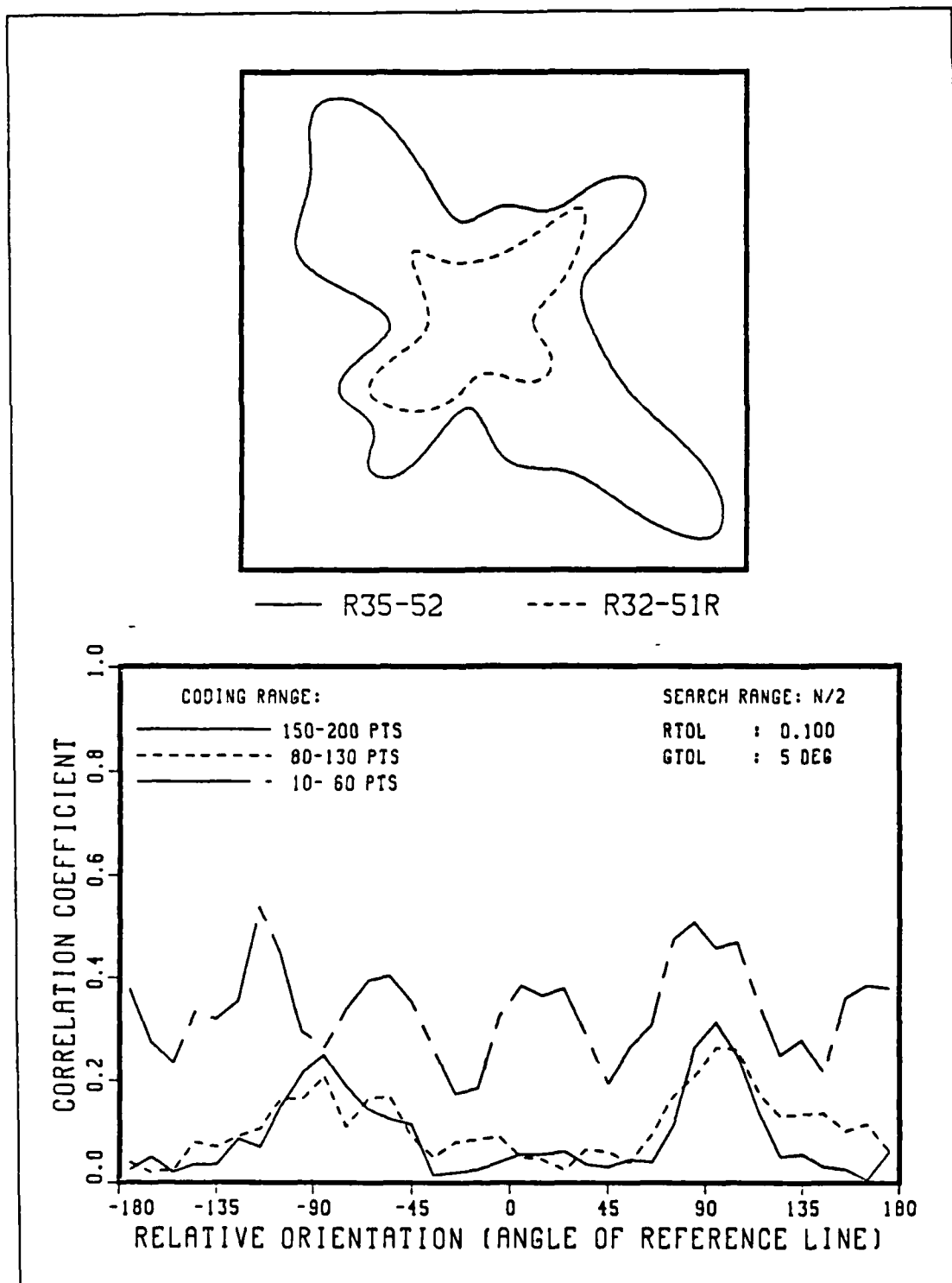


Figure 4.12 Correlation Between R35-52 and R32-51r

The amount of correlation is affected by the noise threshold set by the secondary specifications. If the tolerances on these specifications (ie. GTOL and RTOL) are increased, the peak correlation can be seen to increase from about 30% to 50% (Figure 4.13). Inevitably, the amount of false matches increases too.

Figures 4.14 to 4.20 provide further examples for different sets of shapes. The reference shape becomes progressively 'smoother'. The general level of correlation is higher for these figures than for the previous set. This is due to the general symmetry and gross similarity between these shapes. Figure 4.20 provides the extreme case where the test shape is almost circular. Because of the symmetry, the correlation at all orientation is nearly constant. Also, since there is marked similarity between the test and reference shapes, this level of correlation is also very high. The reader may wonder about the ability of the algorithm to distinguish between very smooth shapes such as ellipses. This is further discussed under the section on Discrimination below.

2. Partial Matching

Figure 4.21 shows the ability of the algorithm to detect partial matches. Except for the lowest coding range, the results show a distinct correlation peak at zero relative orientation. The multiple peaks in the lowest coding range is due to the general similarity of shorter segments compared to longer segments. Figure 4.22 is a plot of the correlated points (for the 150 to 200 coding range). It shows clearly the segment of partial match. Also, it shows that the correlated points at the other orientations are scattered across the boundary. In obtaining the value of correlation, the algorithm simply sums up the number of correlated points at each orientation.

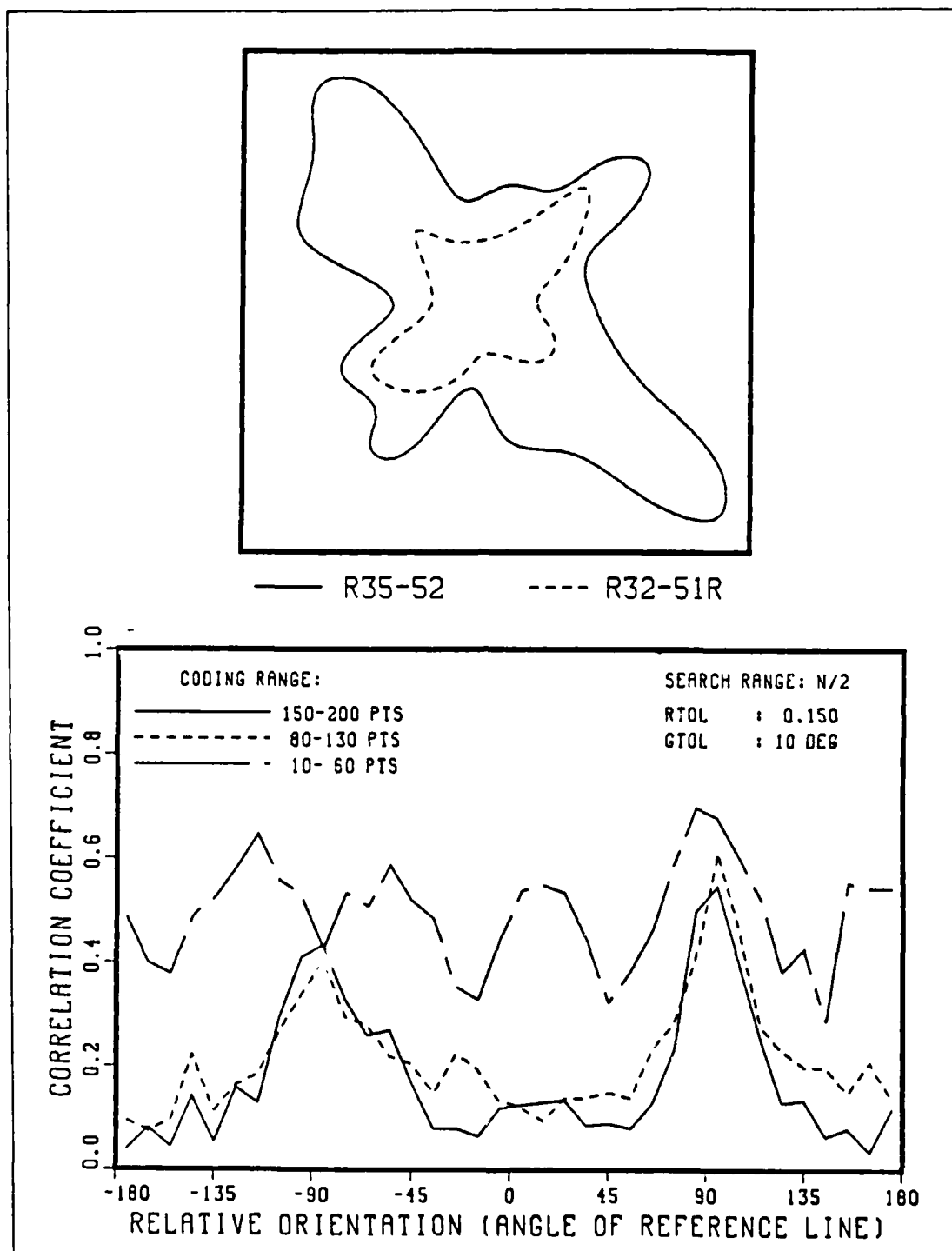


Figure 4.13 Correlation Between R35-52 and R32-51r
Using Relaxed Tolerances

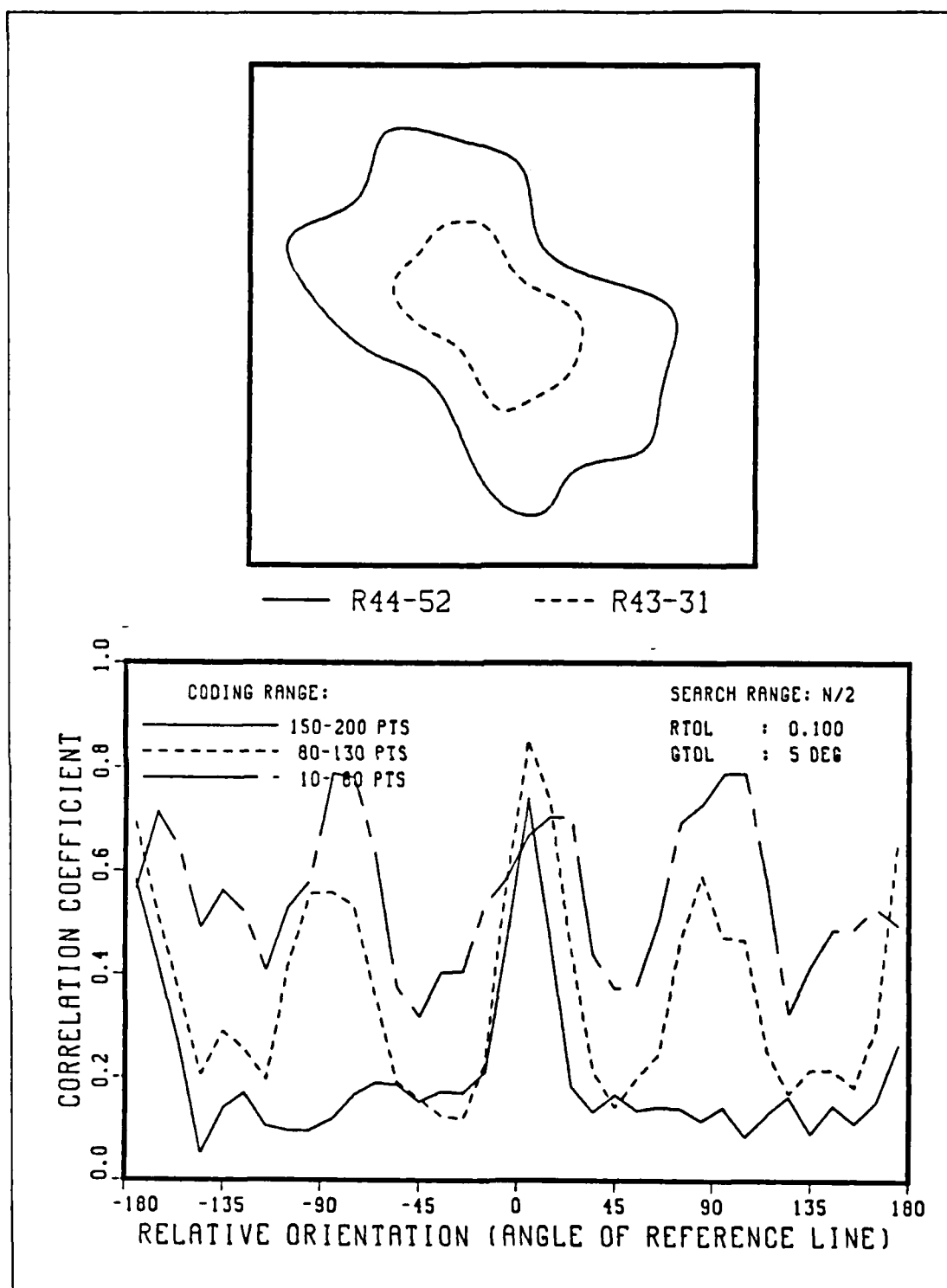


Figure 4.14 Correlation Between R44-52 and R43-31

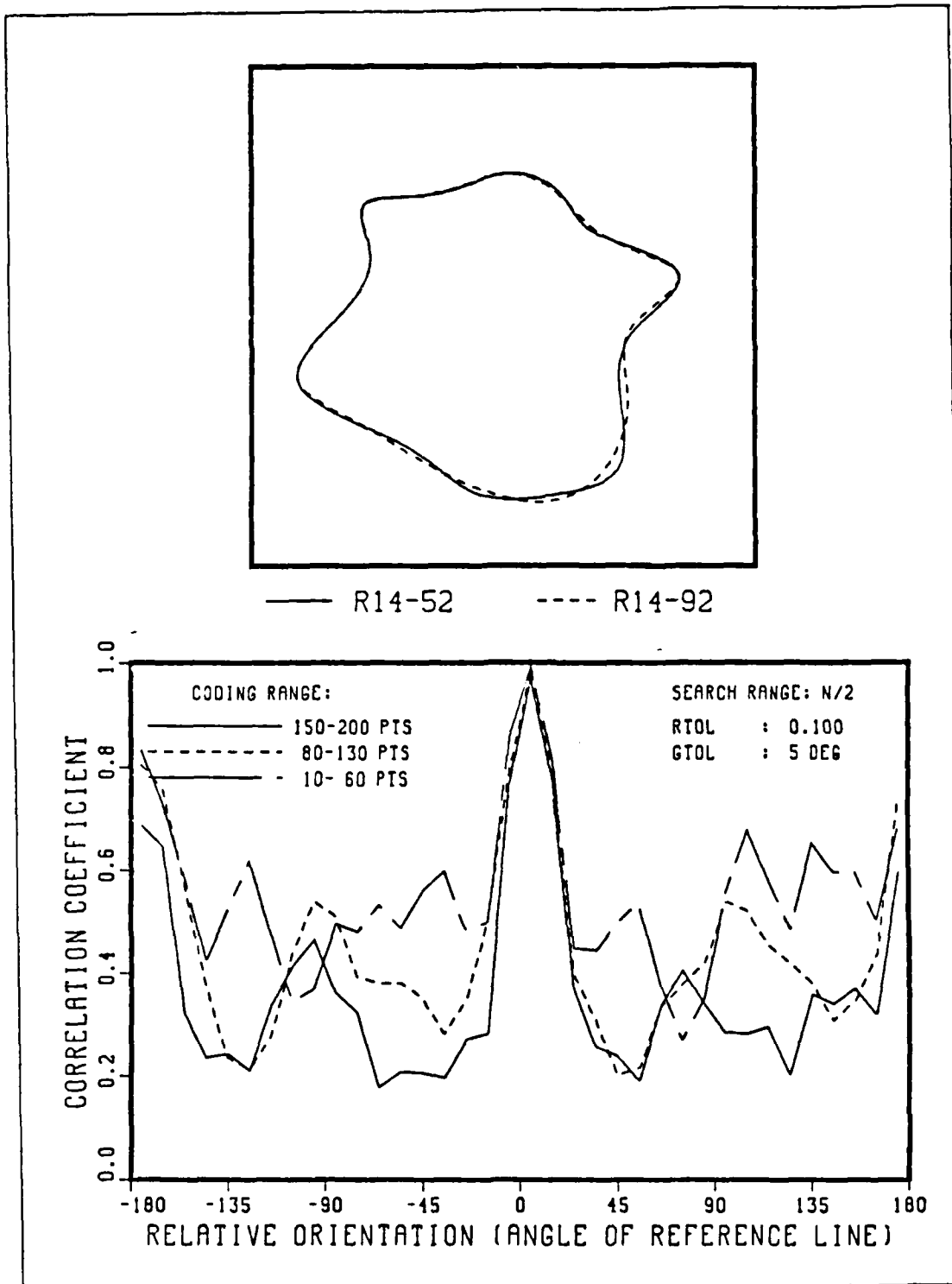


Figure 4.15 Correlation Between R14-52 and R14-92

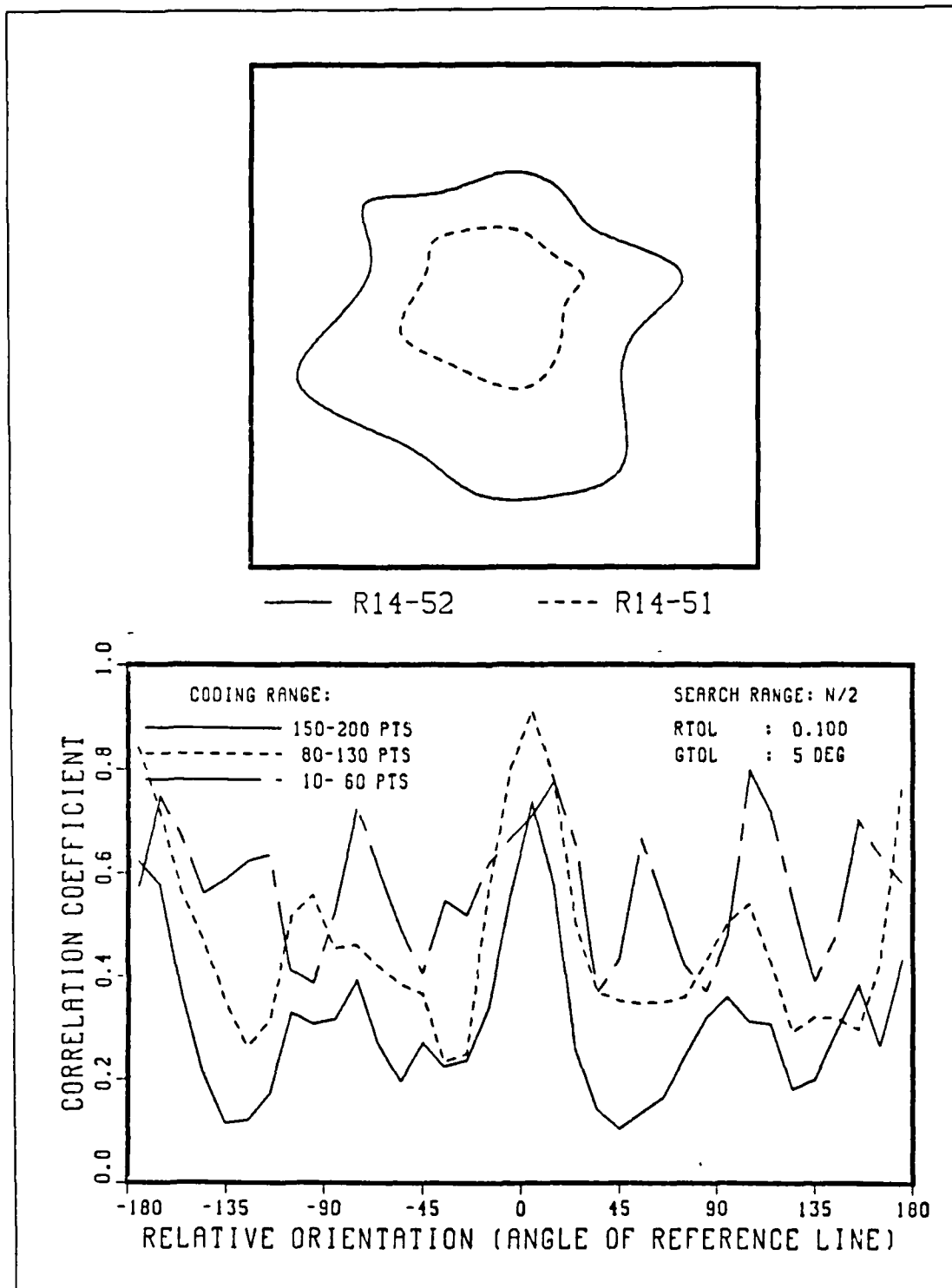


Figure 4.16 Correlation Between R14-52 and R14-51

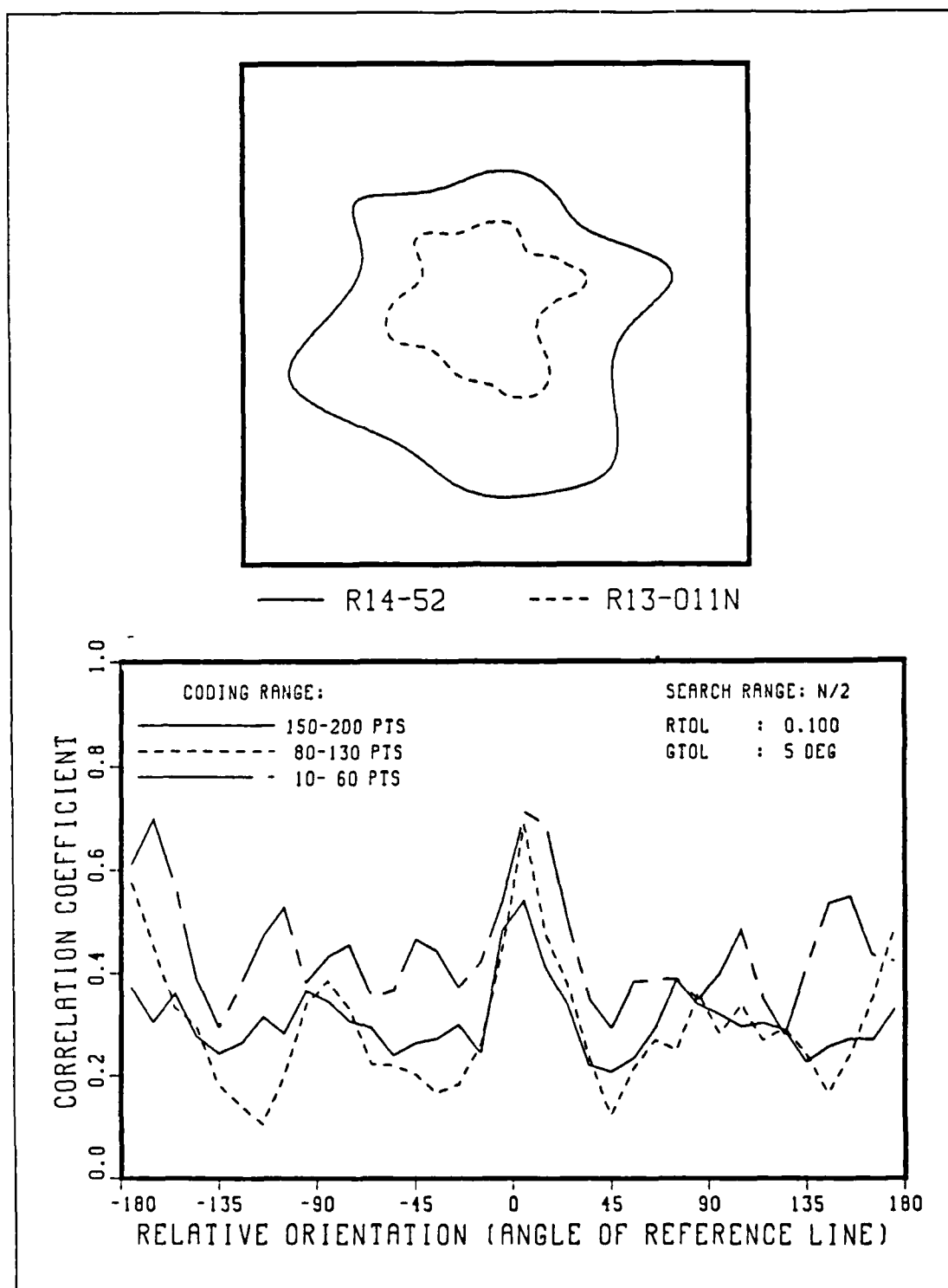


Figure 4.17 Correlation Between R14-52 and R13-011n

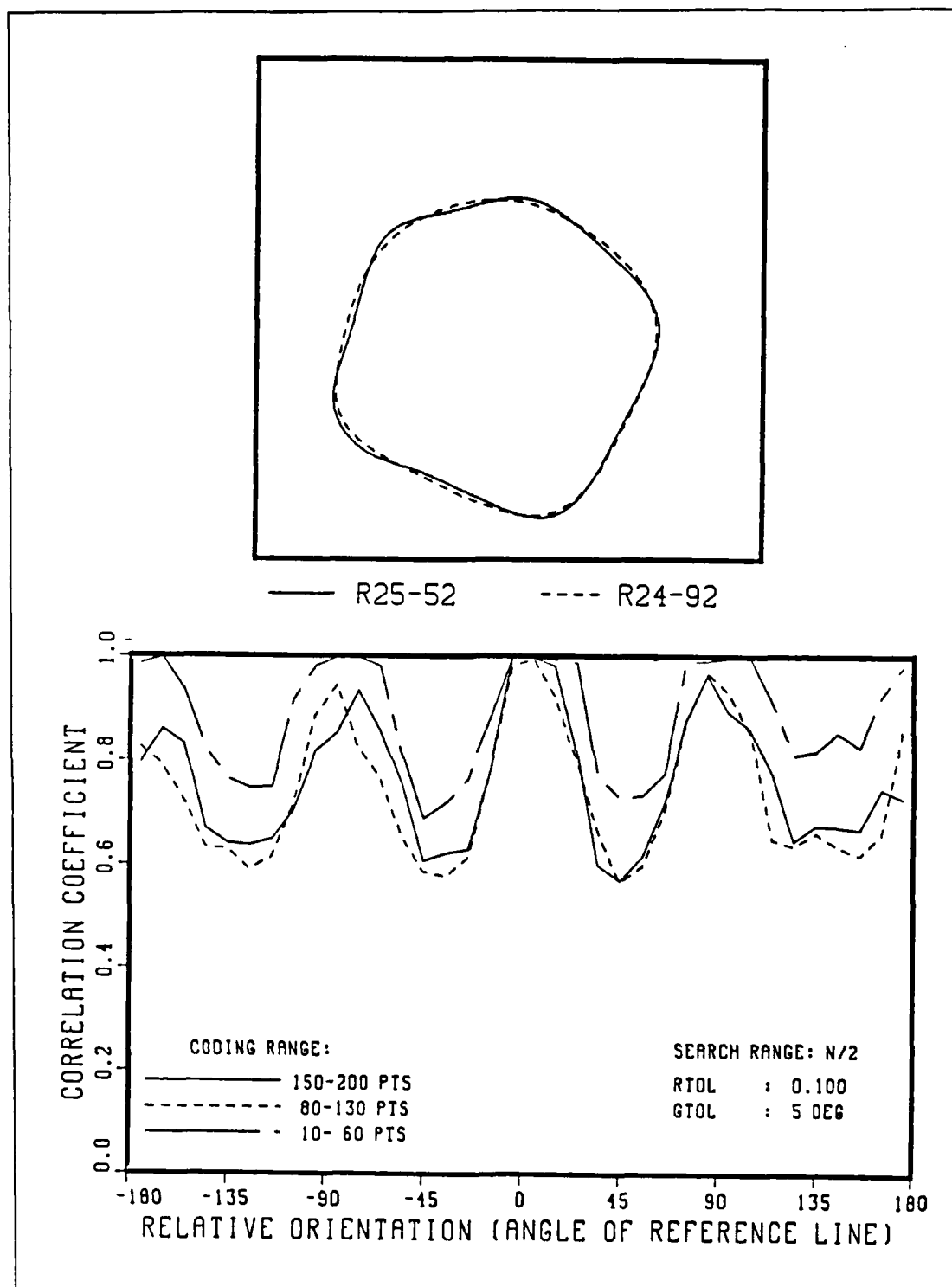


Figure 4.18 Correlation Between R25-52 and R24-92

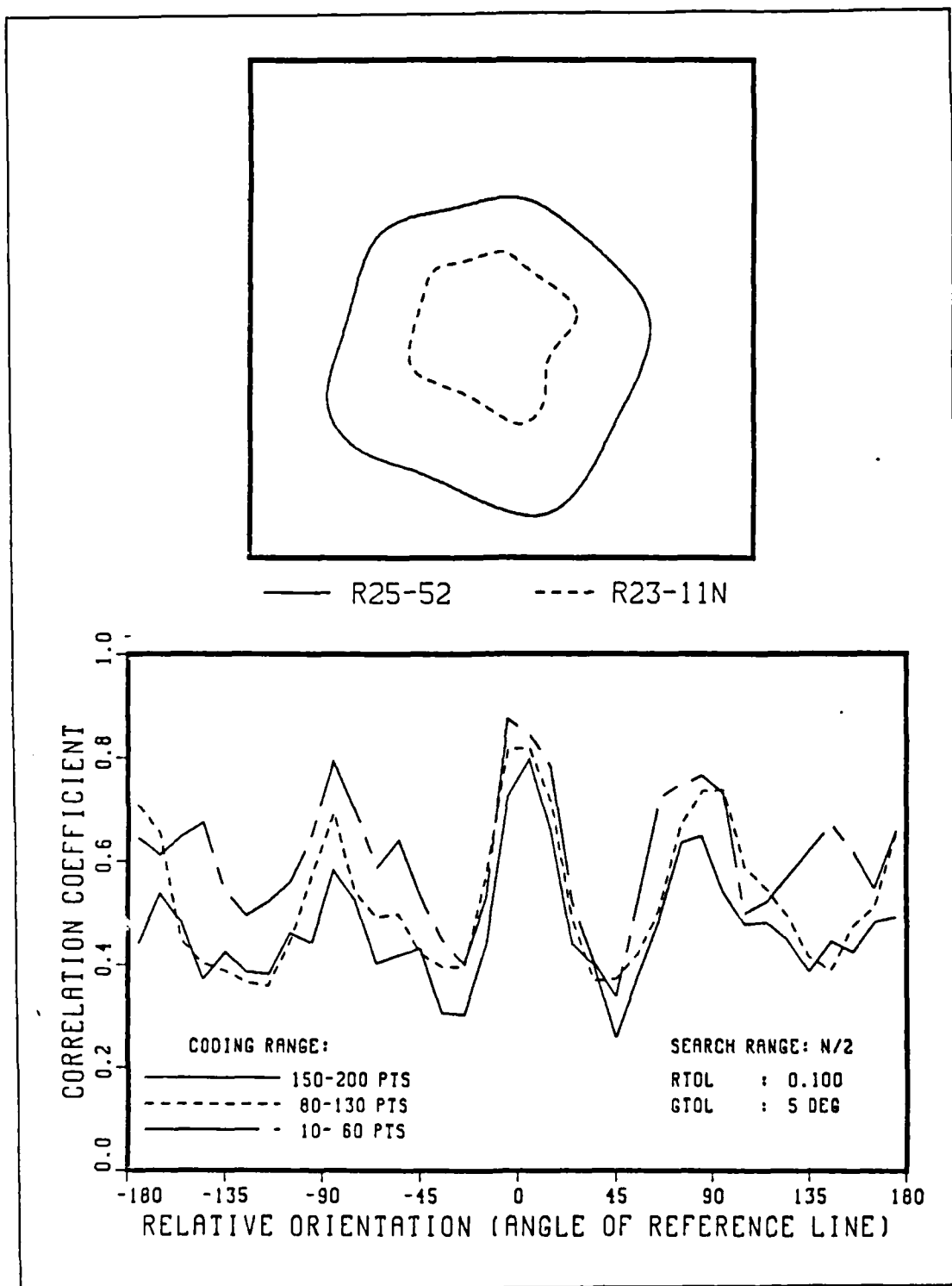


Figure 4.19 Correlation Between R25-52 and R23-11n

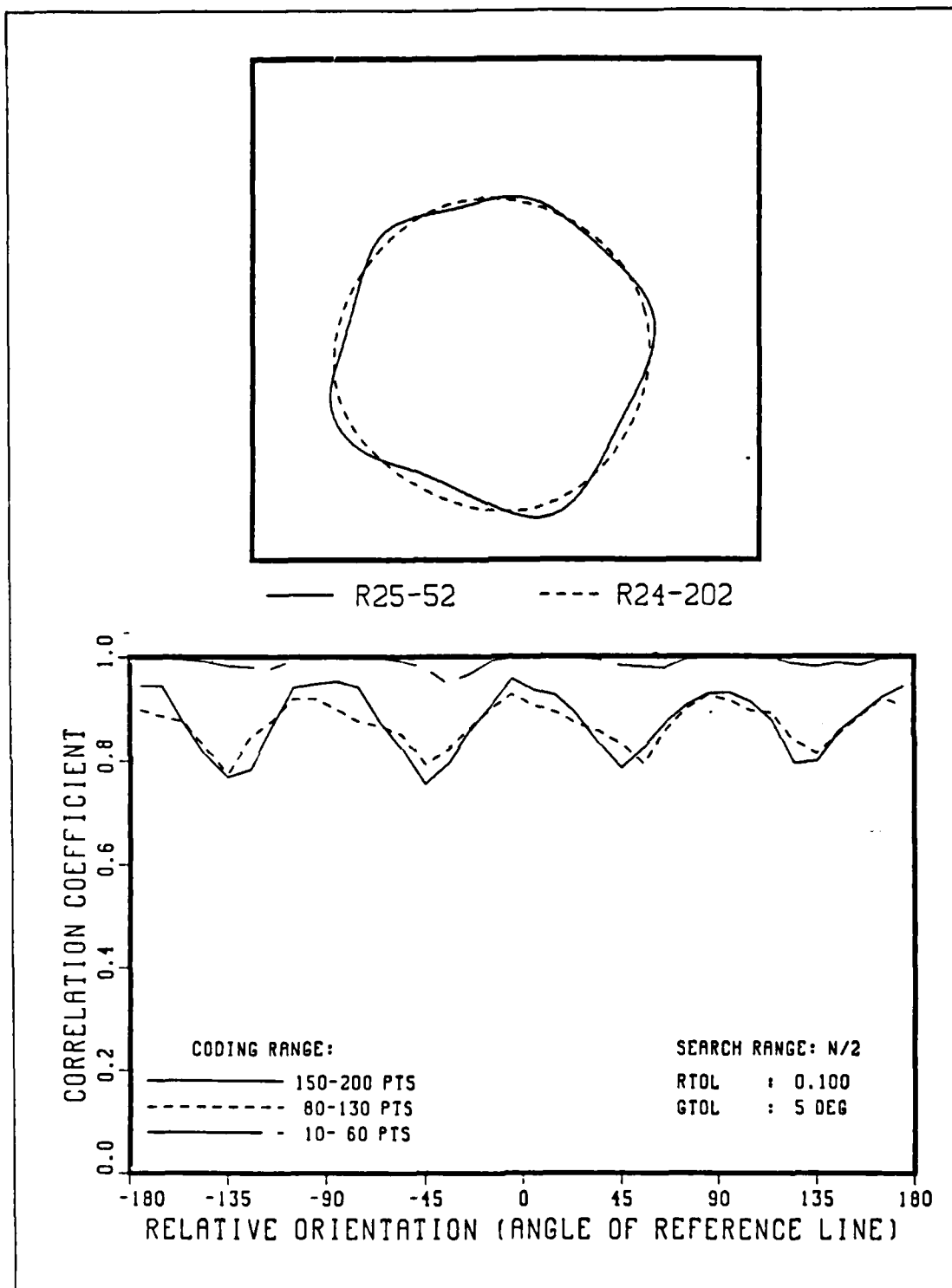


Figure 4.20 Correlation Between R25-52 and R24-202

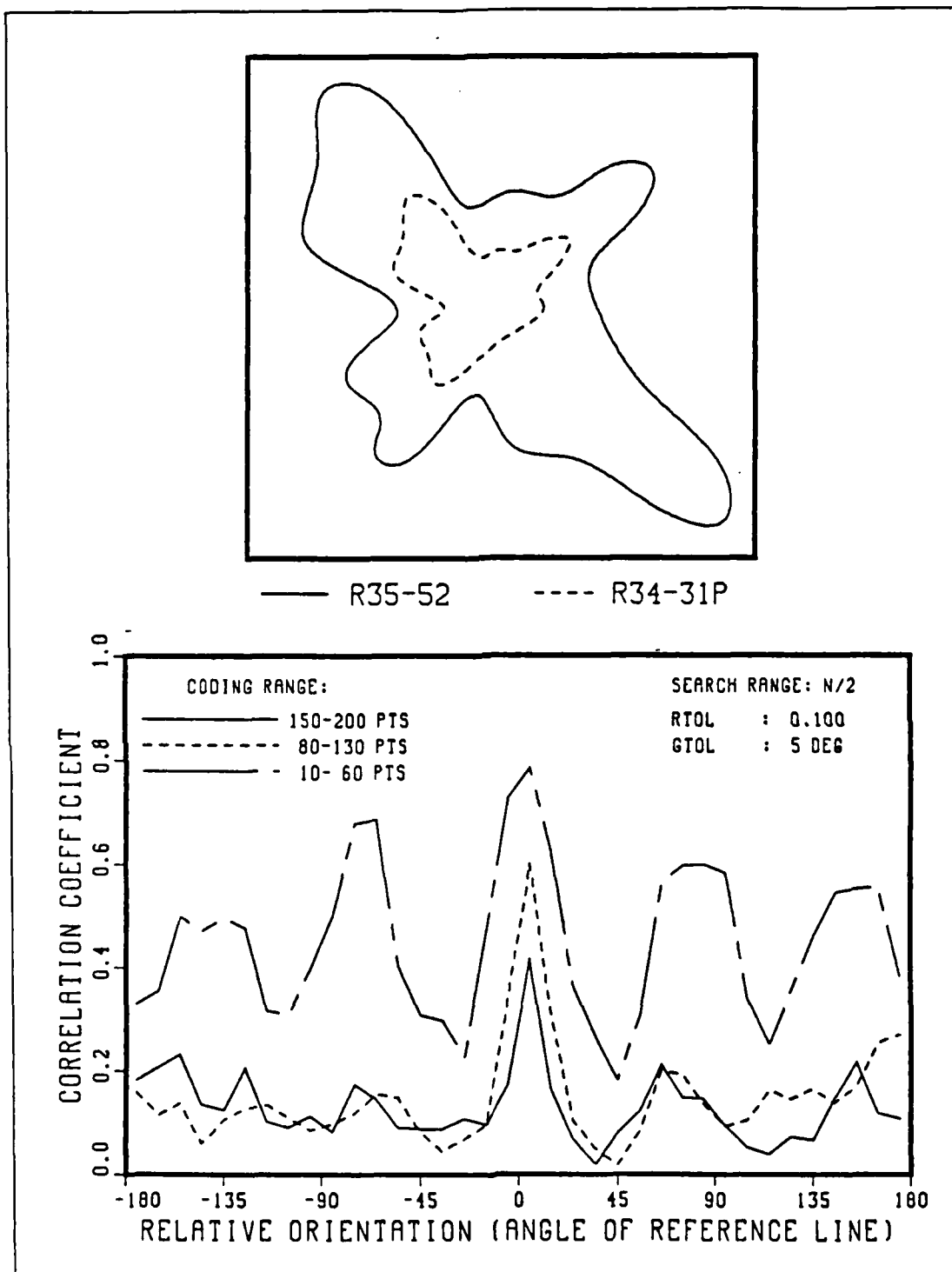


Figure 4.21 Correlation Between R35-52 and R34-31p

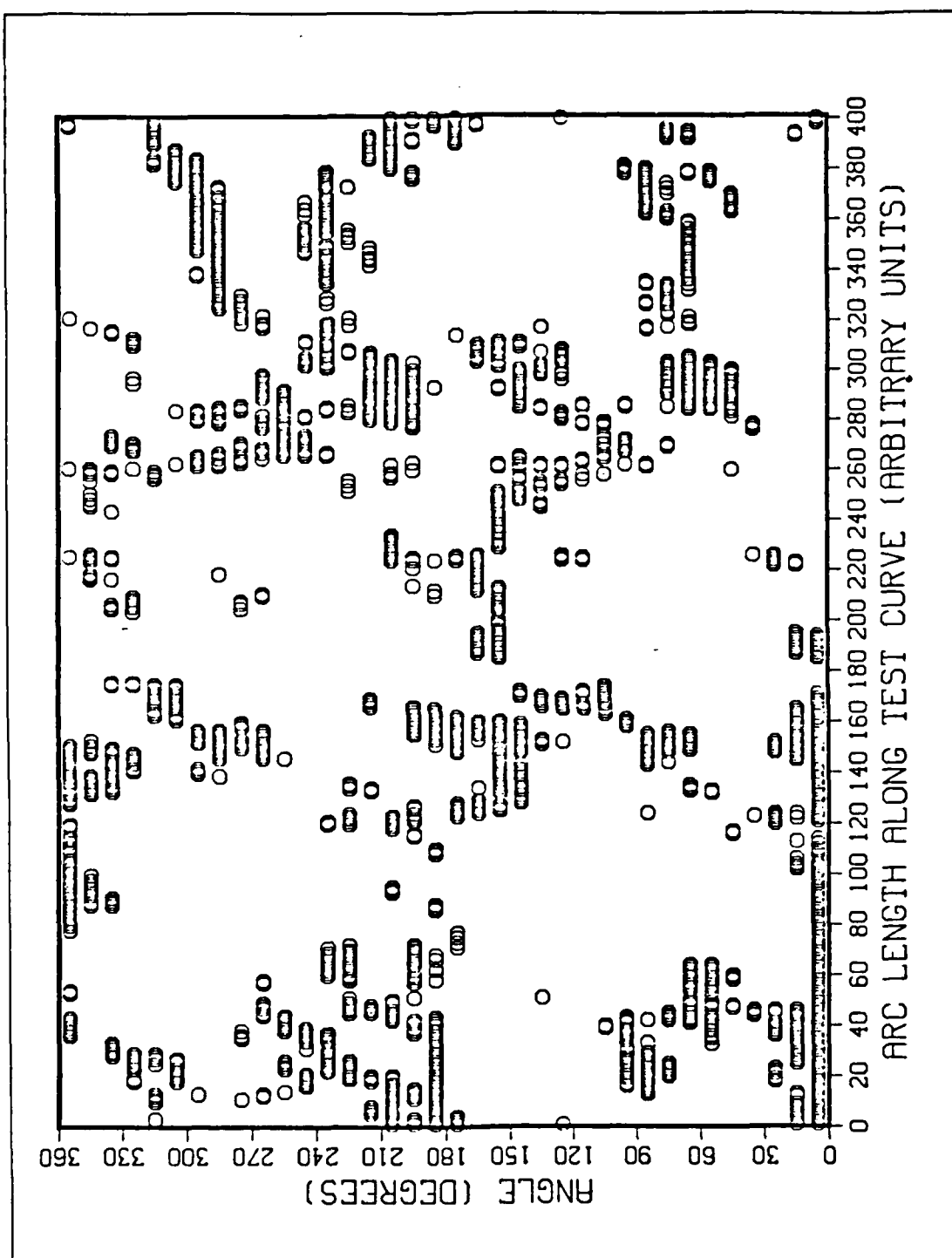


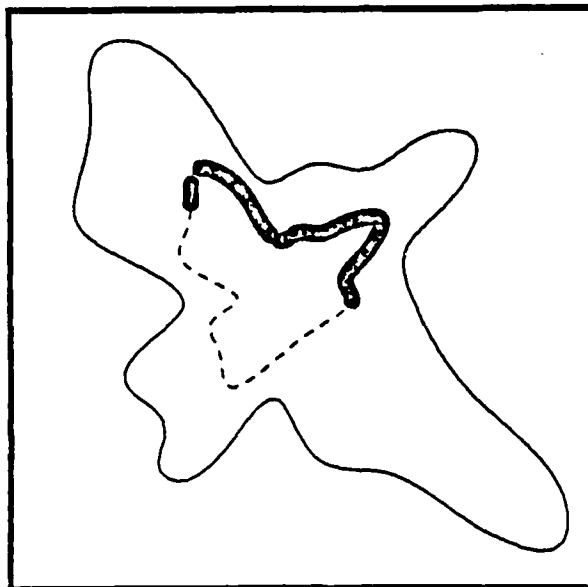
Figure 4.22 Detailed Correlation Between R35-52 and R34-31p for Coding Range 150 to 200

Figure 4.23 indicates the location of the matched segments for two coding ranges. The ability of the algorithm to correctly locate the matched segments is clearly illustrated. The two diagrams also show clearly the effects of 'end losses'. At the 150 to 200 coding range, the 'look forward' section is much longer than for the 10 to 60 range. Consequently, the higher the loss of matched points at the forward end. As mentioned before, this loss could be minimised by modifying the coding and matching algorithm to look in both directions.

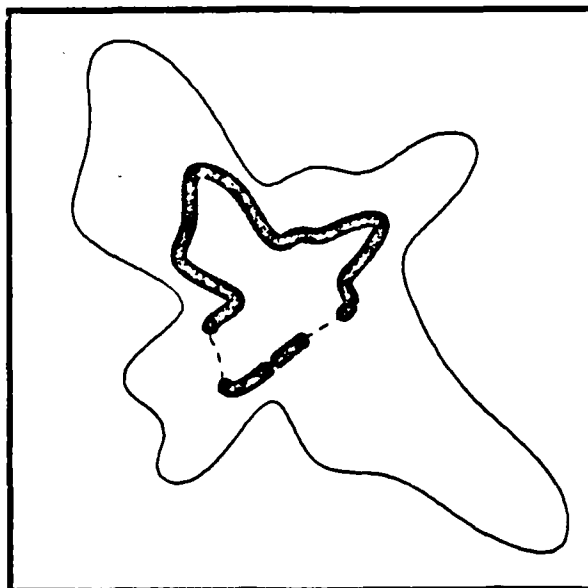
Figures 4.24 to 4.26 show the effect of noise on partial matching. As before, the peak correlation decreases with noise while the off-peak level remains relatively constant. Note that the coding range 150 to 200 produces almost zero correlation. This is not surprising since the reference shape boundary has only 200 data points. At this coding range, almost the entire boundary is being coded at each point! This illustrates clearly the relationship between the coding range and the 'local' characteristics in the coding. For partial match applications, it is essential that the coding range be restricted to a short section of the boundary. Figure 4.27 shows the location of the partial match for the relative orientation -75 degrees.

Figures 4.28 shows the matching of a small section of a 'wing' to the reference shape R32-31r. A good match is found at about -75 degrees. Figure 4.29 shows the reverse situation, where the reference shape is matched against the given wing. Possible matches are located at about 95 degrees and -105 degrees. The matched segment is indicated in Figure 4.30 (for orientation 95 degrees). These segments agree with our visual observation.

Figures 4.31 to 4.32 provide more examples of partial matches. Note that in all these, the location of the



(B) 150 TO 200 RANGE



(A) 10 TO 60 RANGE

**Figure 4.23 Matched Segment Between R35-52 and R34-3lp
at Zero Degree Relative Orientation**

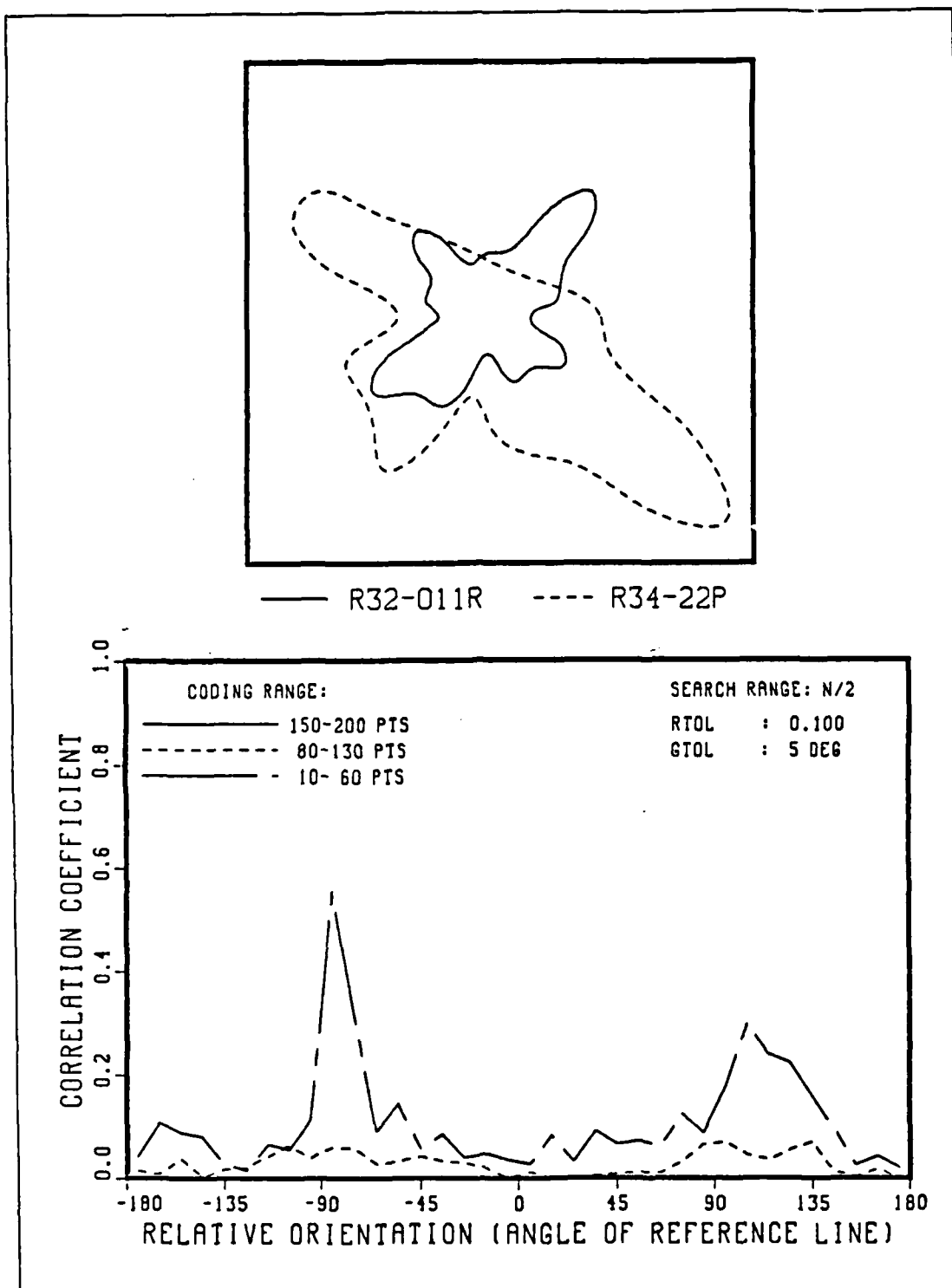


Figure 4.24 Correlation Between R32-011r and R34-22p

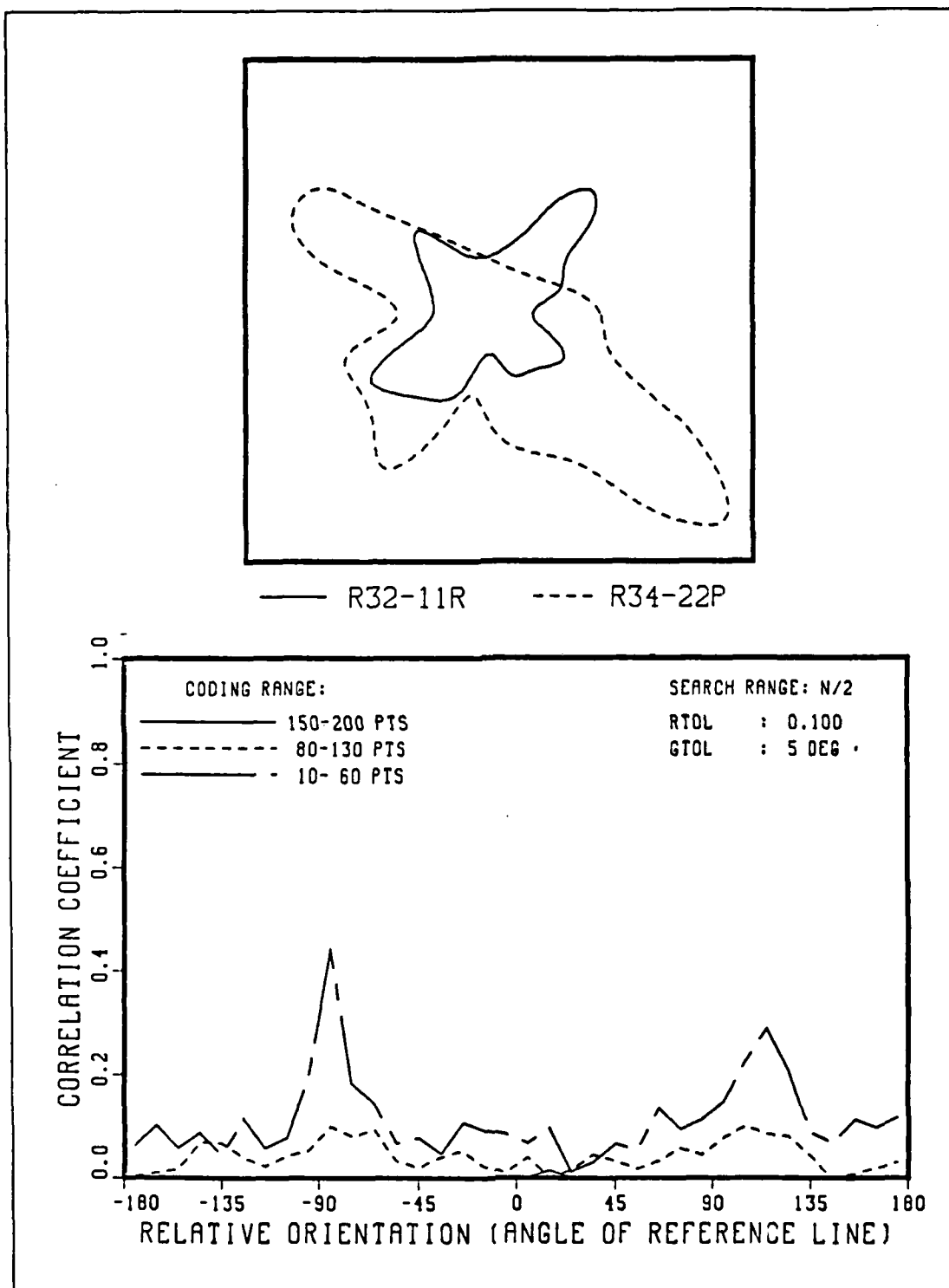


Figure 4.25 Correlation Between R32-11r and R34-22p

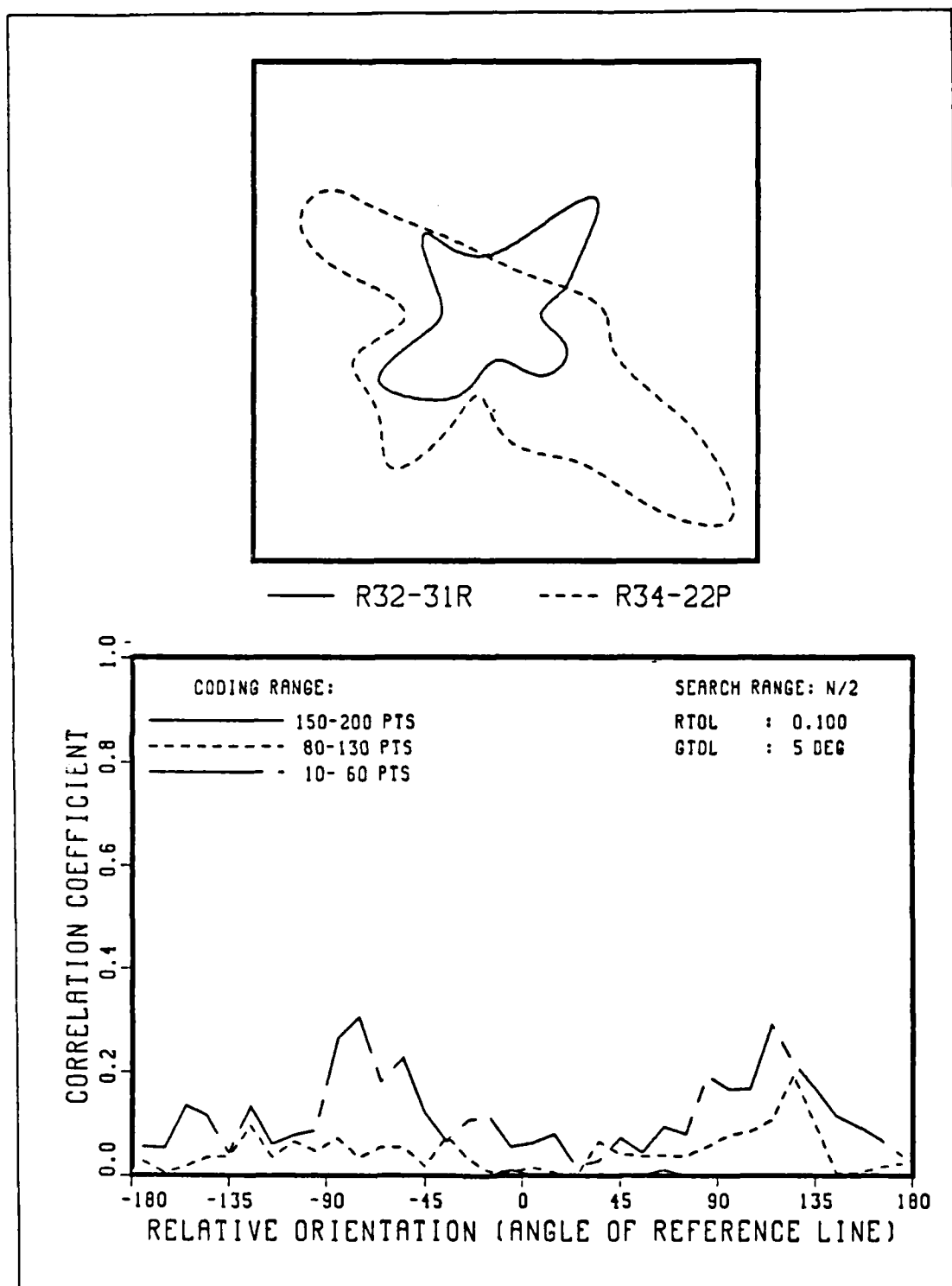
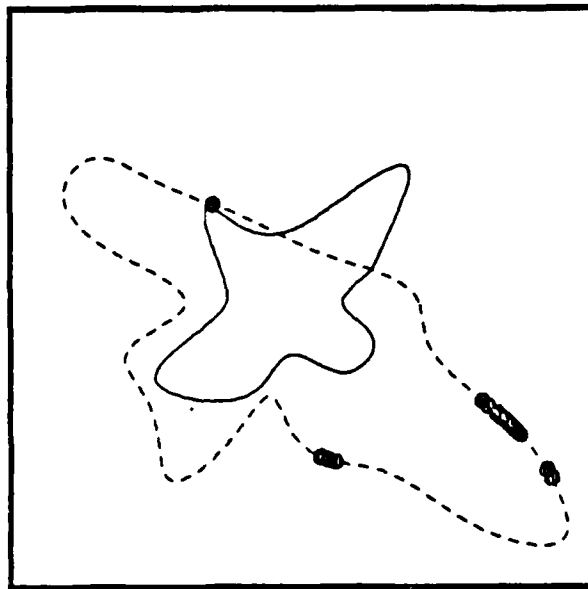
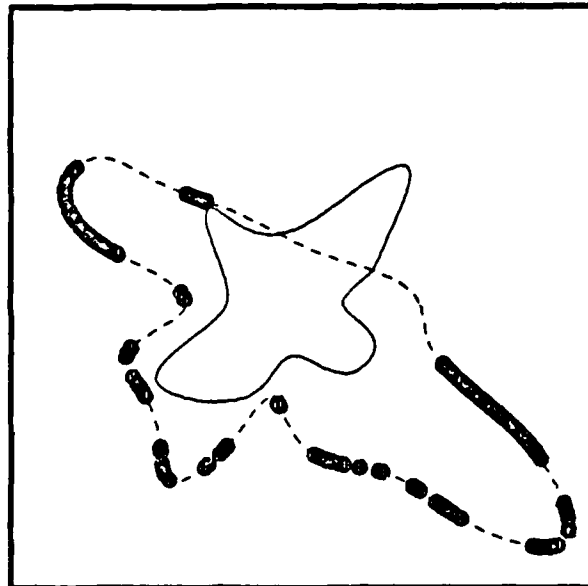


Figure 4.26 Correlation Between R32-31r and R34-22p



(B) 80 TO 130 RANGE



(A) 10 TO 60 RANGE

Figure 4.27 Matched Segments Between R32-31r and R34-22p
at -75 Degrees Relative Orientation

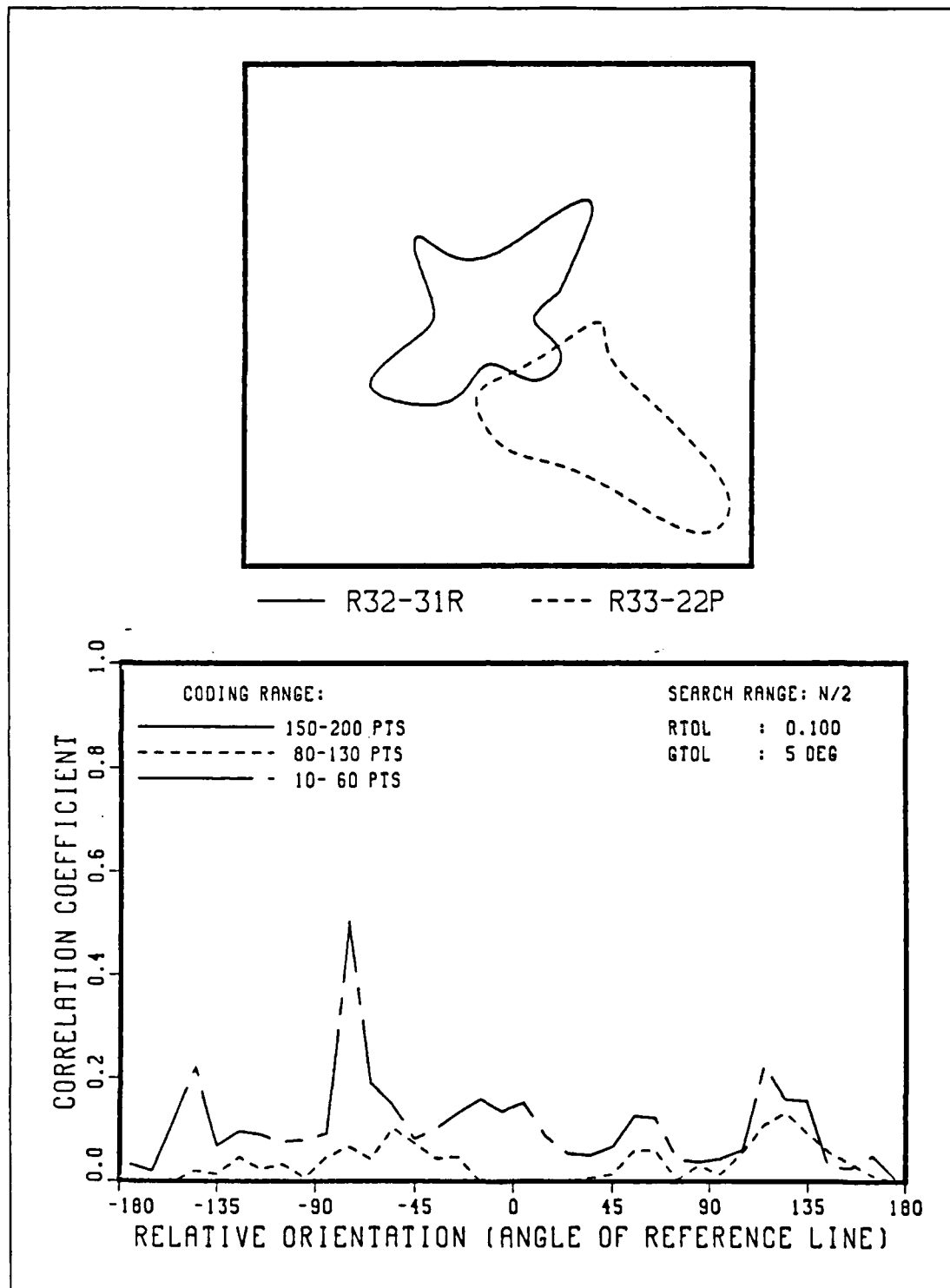


Figure 4.28 Correlation Between R32-31r and R33-22p

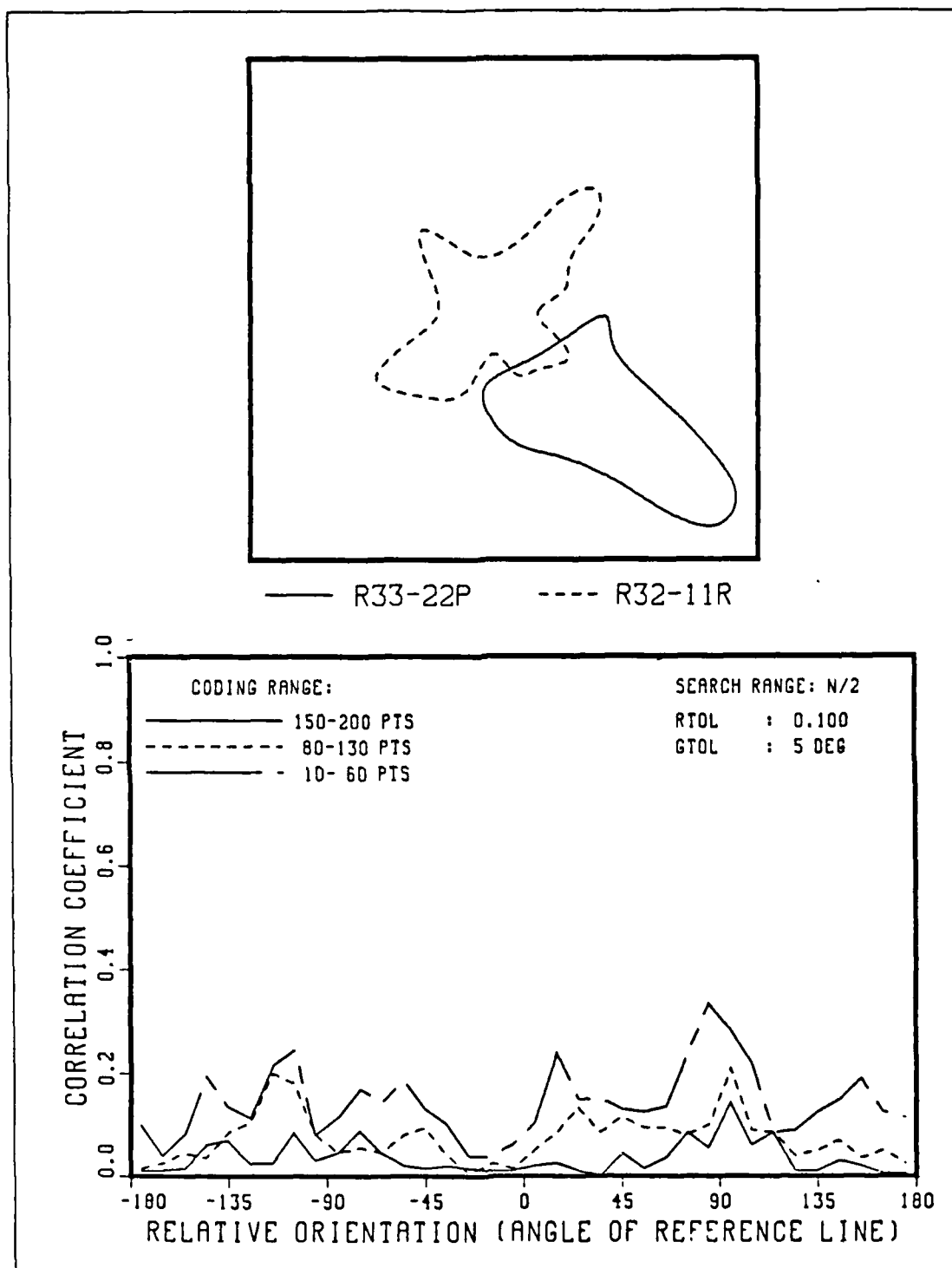
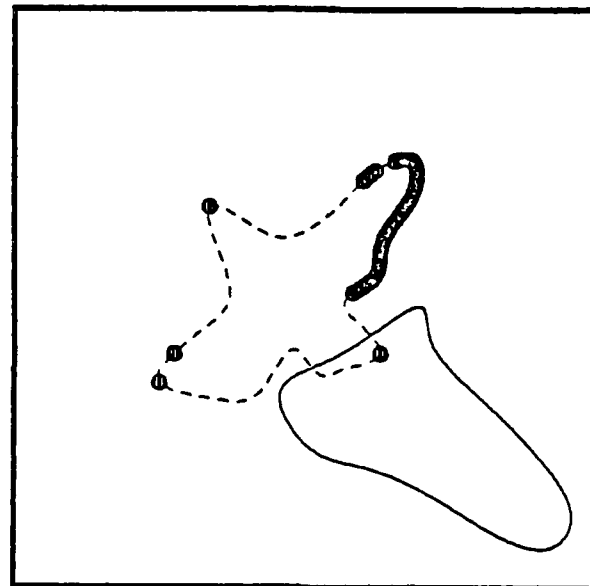
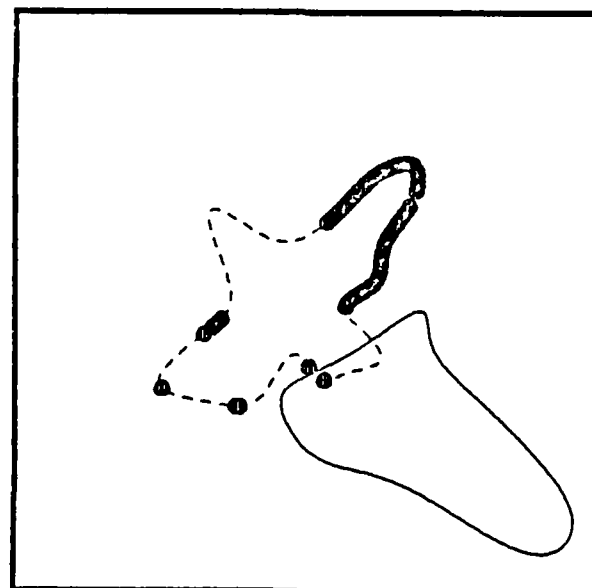


Figure 4.29 Correlation Between R33-22p and R32-11r



(B) 80 TO 130 RANGE



(A) 10 TO 60 RANGE

Figure 4.30 Matched Segments Between R33-22p and R32-11r
at 95 Degrees Relative Orientation

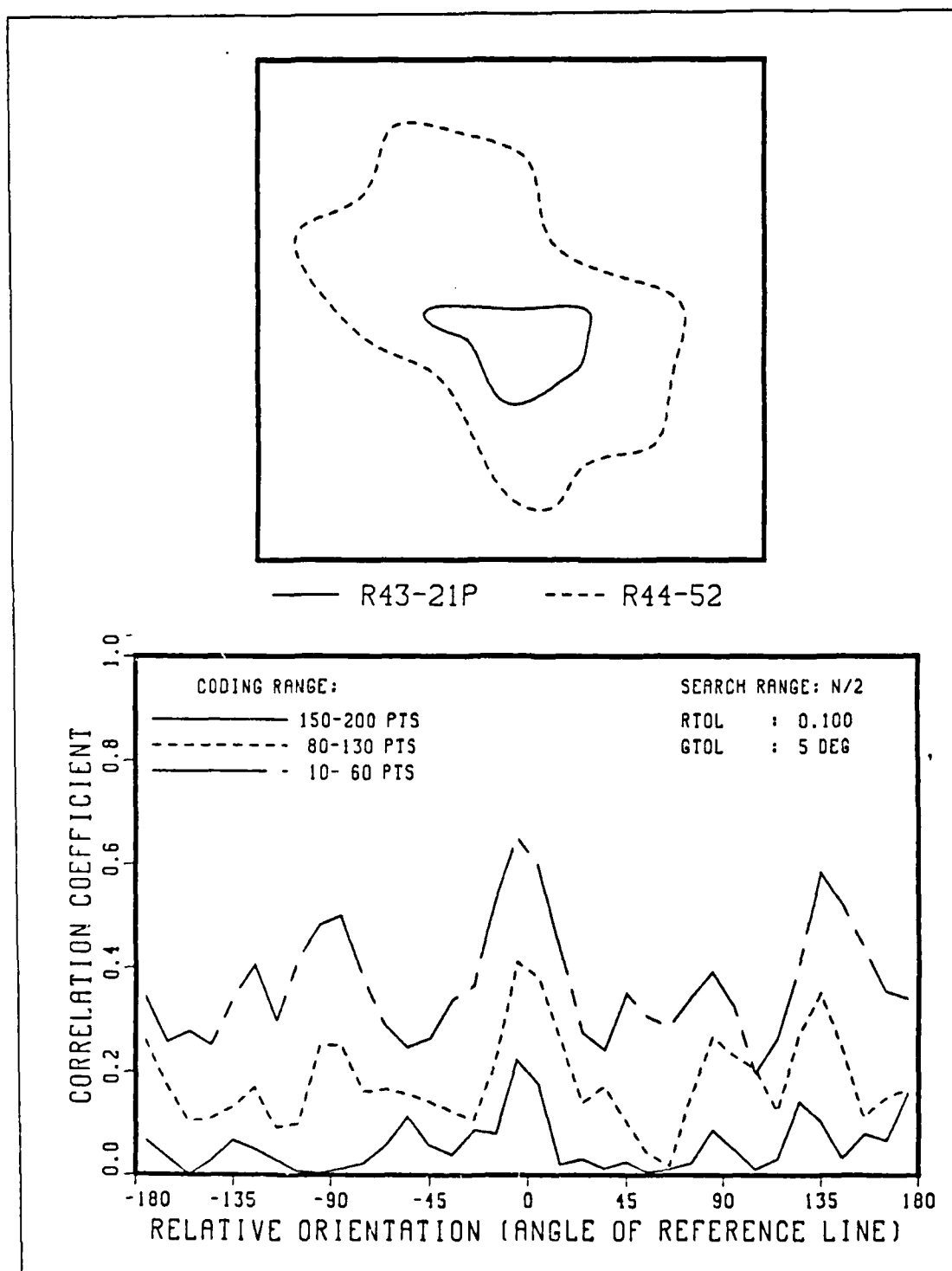


Figure 4.31 Correlation Between R43-21p and R44-52

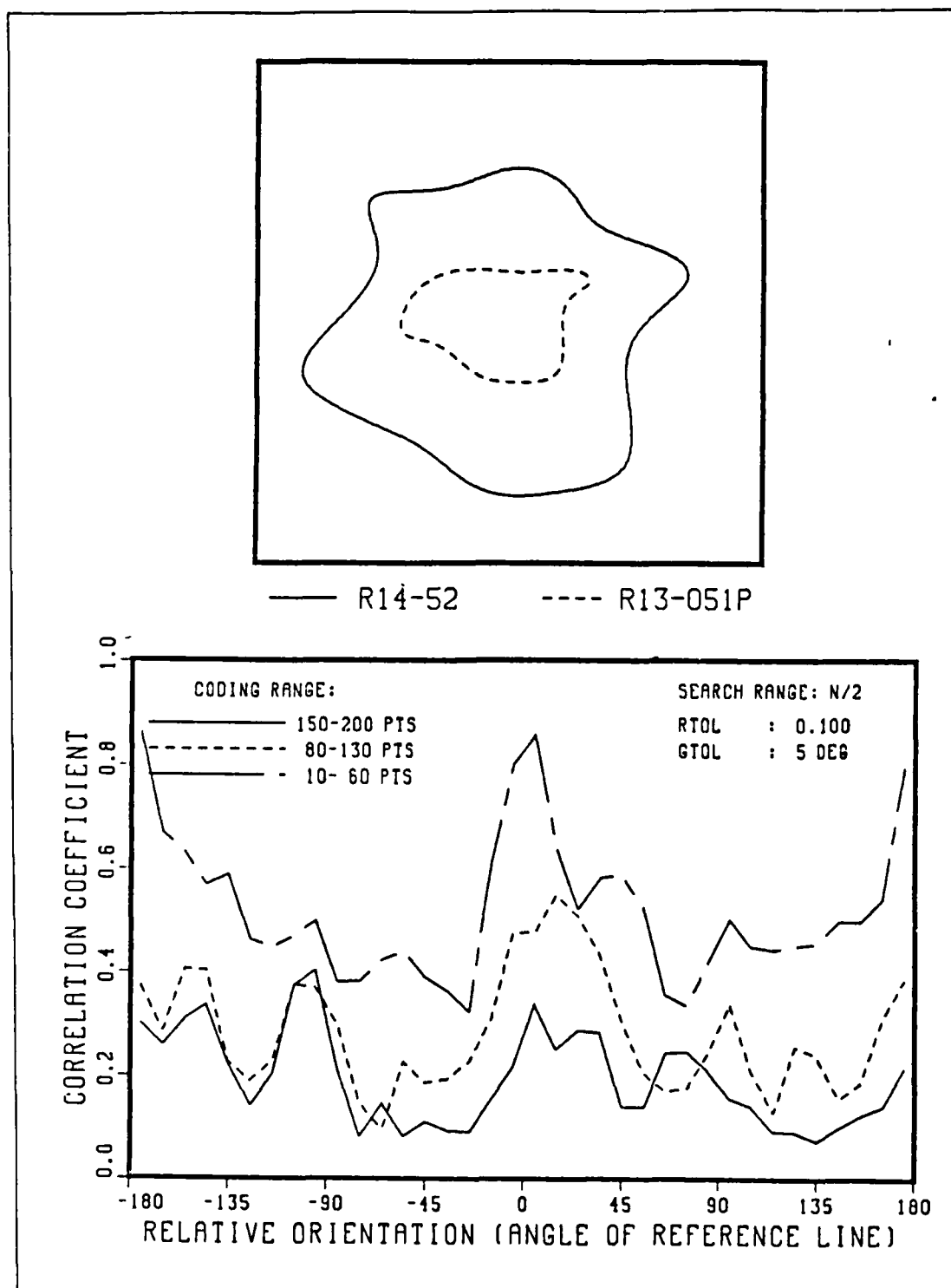


Figure 4.32 Correlation Between R14-52 and R13-051p

peak correlation is correctly obtained. However, because of the general symmetry in the shapes, the general level of the correlation (away from the peak) is also significant. If the 'scatter' of correlated points is taken into account, these false matches could possibly be reduced. The simplest way to do this would be to give different weightings to the correlated points depending on whether these are isolated points or are part of a continuous segment.

3. Discrimination Capability

In this final section, we examine the discrimination capability of the algorithm. Figures 4.33 to 4.35 show the low correlation found when matching R35-52 against the other shapes. The next set of examples (Figures 4.36 to 4.39) show the discrimination between 'smoother' class of shapes. There is no prominent peaks in the correlation. However the general level of correlation is significantly higher because of the nature of the shape (smooth with plenty of linear segments). Consider Figure 4.39 for example. The large number of linear segments in both shapes gives rise to the high value of correlation between them.

Figure 4.40 shows the location where partial match is found (at -175 degrees). This figure illustrates the main weakness of this algorithm; it does not check whether the relative positions of the matched segments in both the test and reference shapes are consistently related. In this particular example, different segments in the test shape have obviously been matched to the same segment in the reference shape. To overcome this, one possible solution would be to use some sort of hierarchical matching scheme whereby the matched segments are first arranged according to their lengths and then checked for consistencies; beginning with the longest matched segment, and so on.

The question of the ability to distinguish between highly symmetrical shapes such as ellipses has been raised earlier. Figures 4.41 and 4.42 show how the algorithm

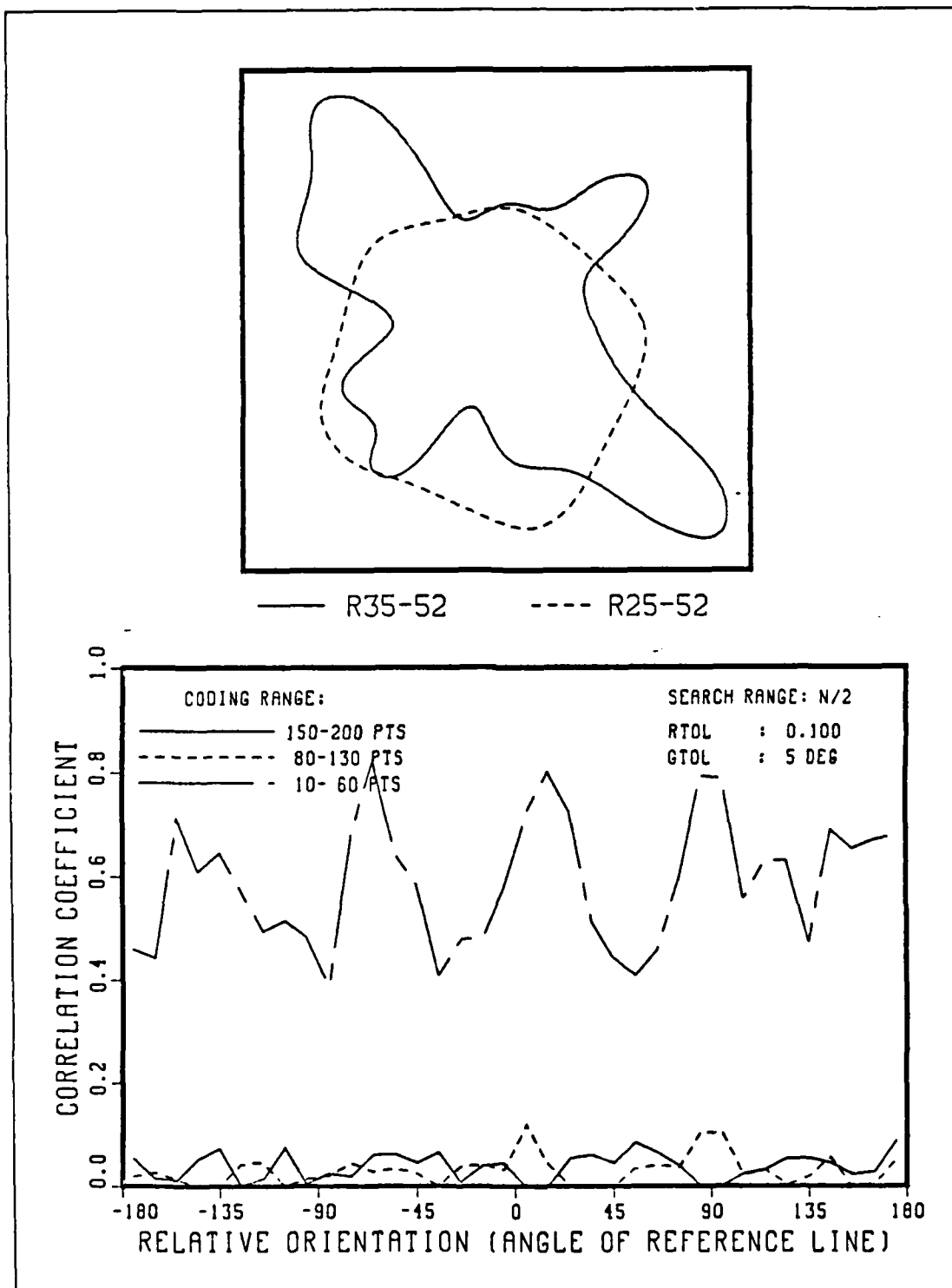


Figure 4.33 Correlation Between R35-52 and R25-52

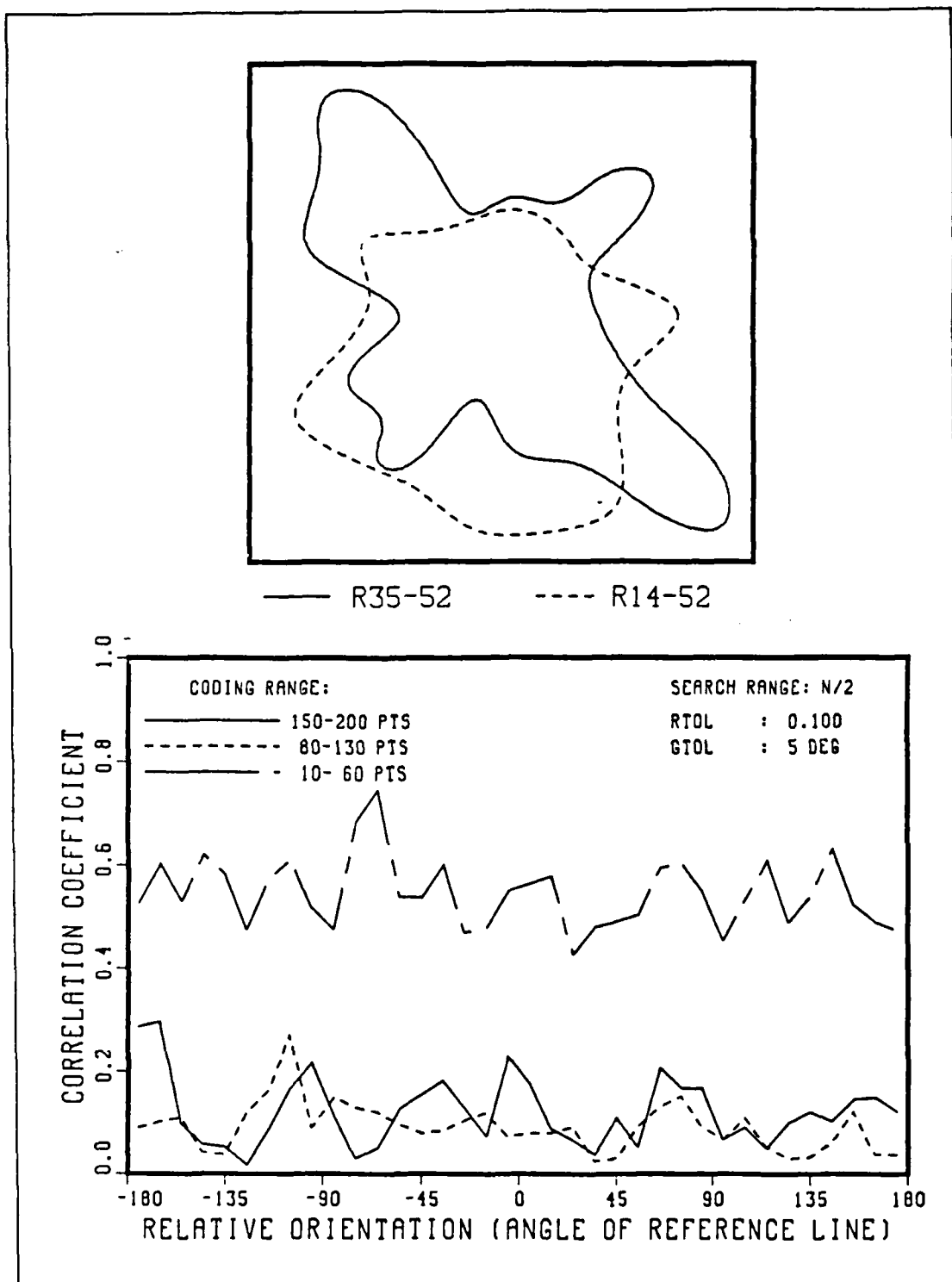


Figure 4.34 Correlation Between R35-52 and R14-52

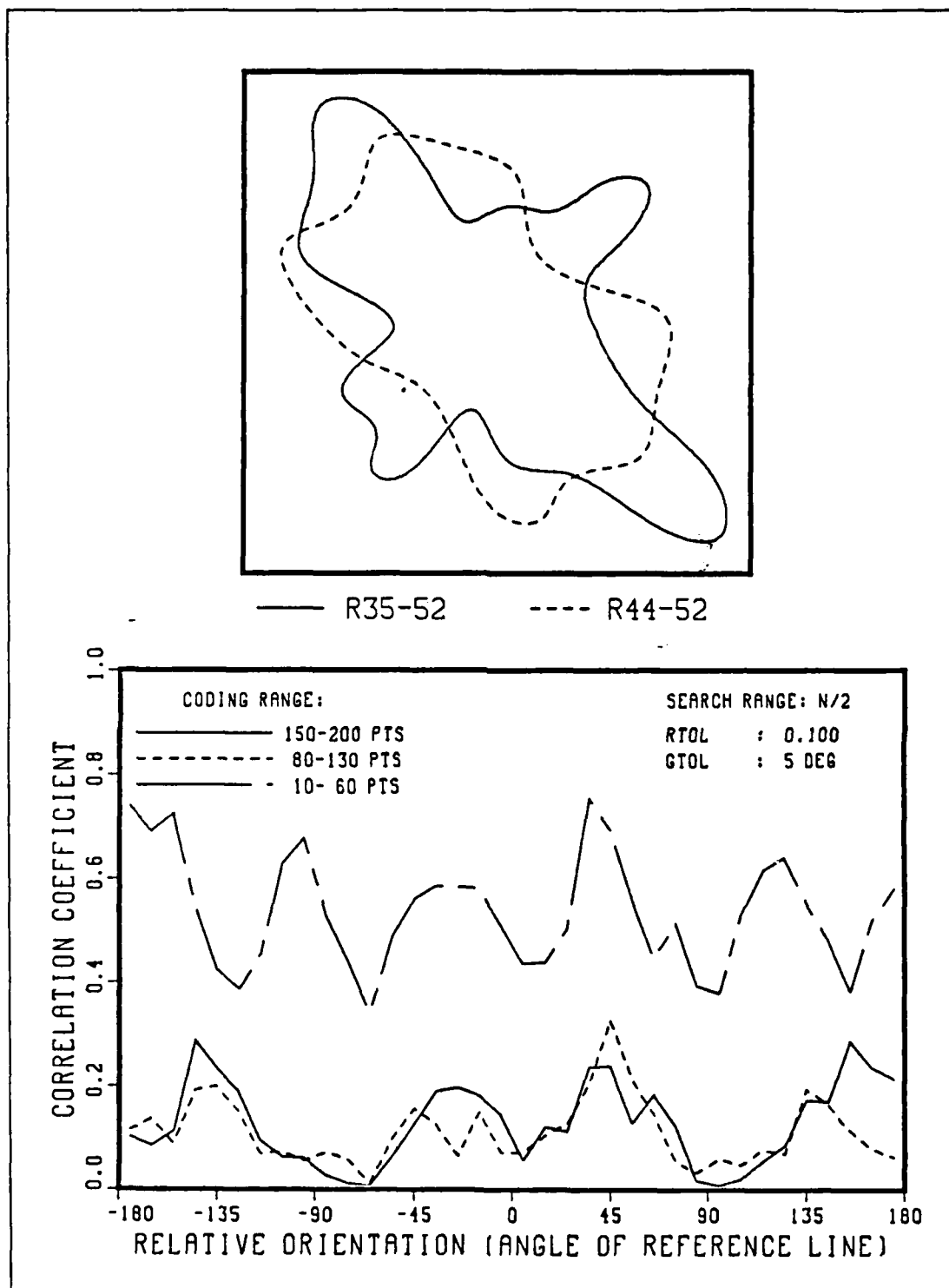


Figure 4.35 Correlation Between R35-52 and R44-52

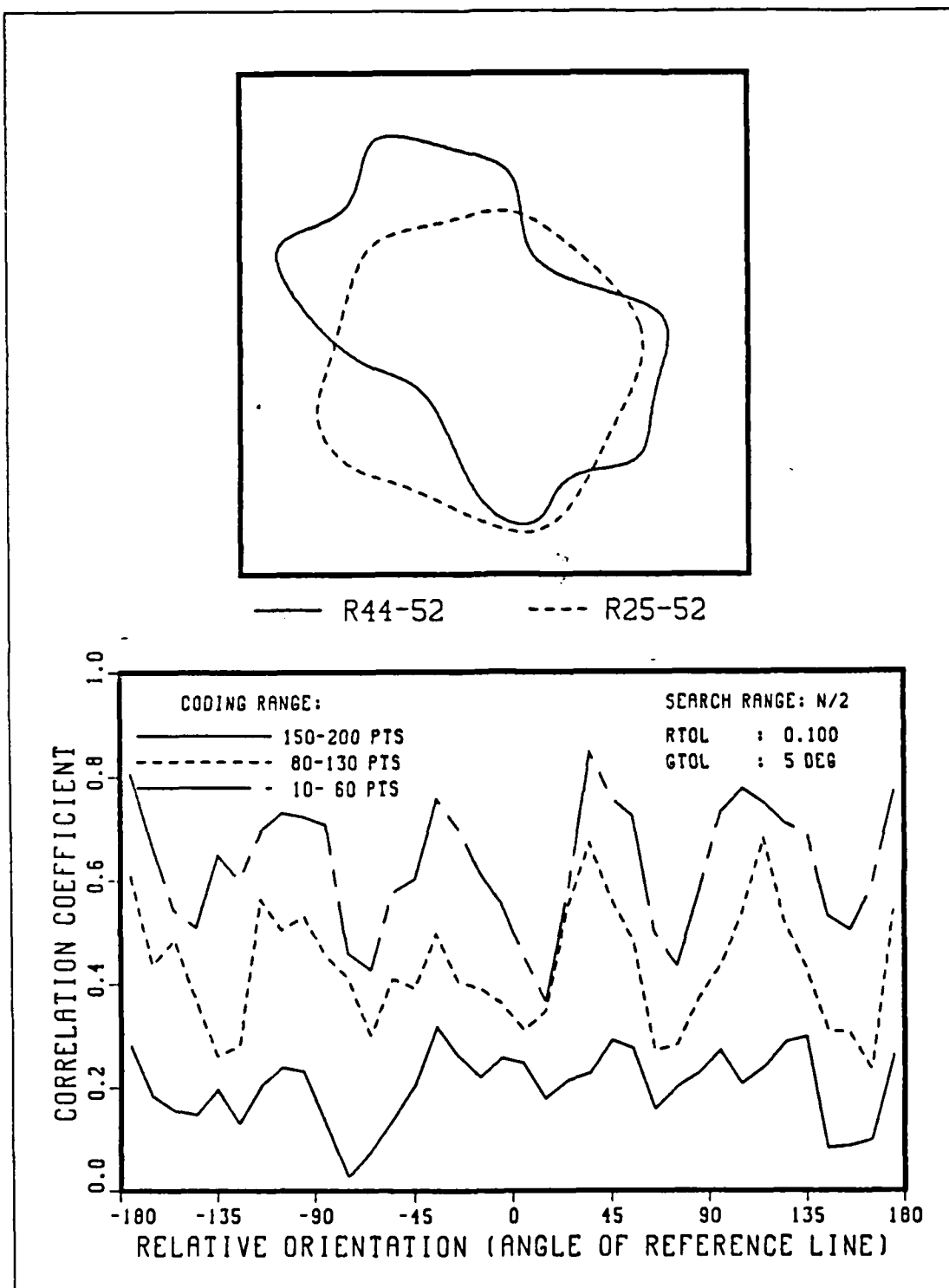


Figure 4.36 Correlation Between R44-52 and R25-52

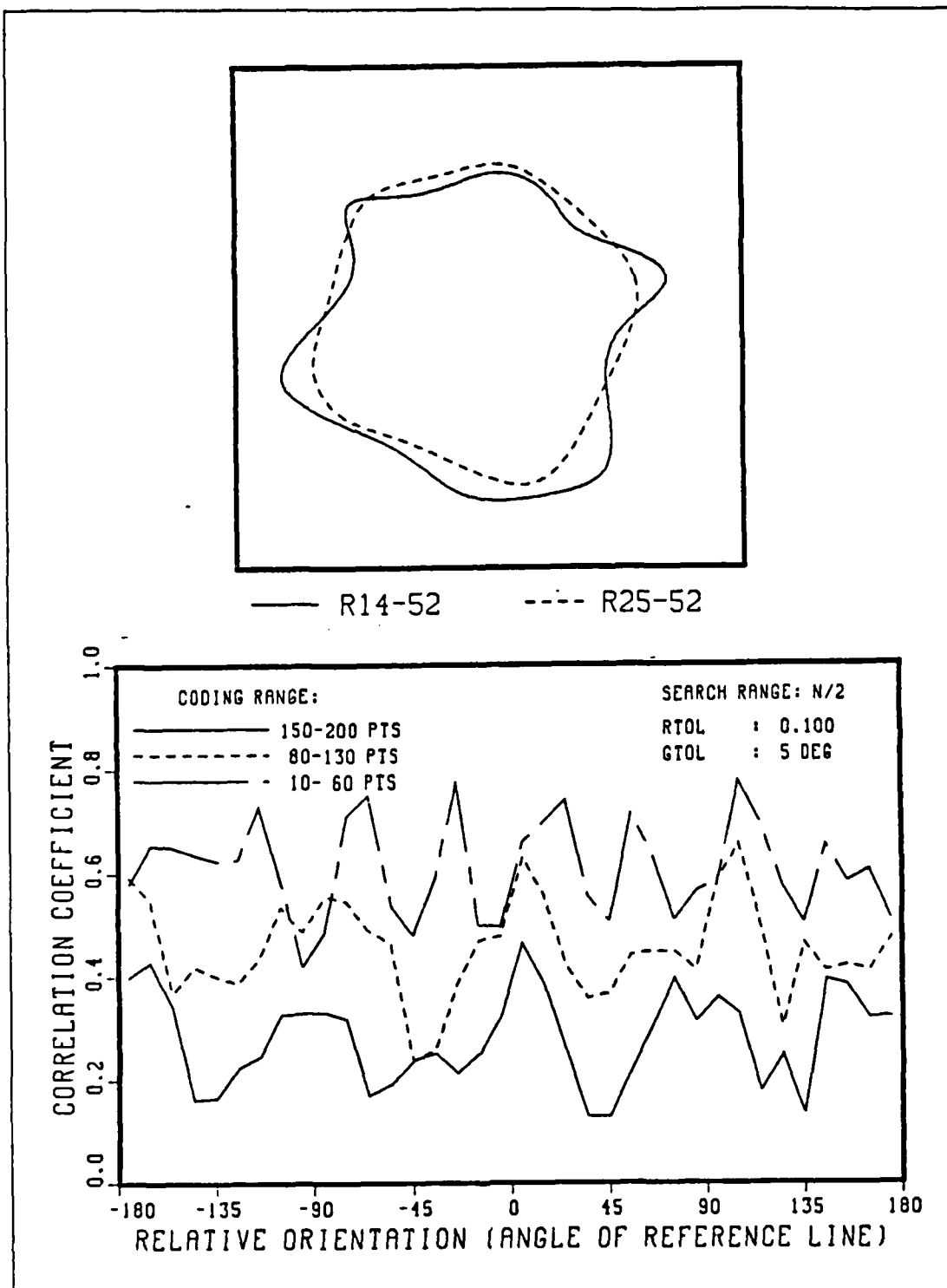


Figure 4.37 Correlation Between R14-52 and R25-52

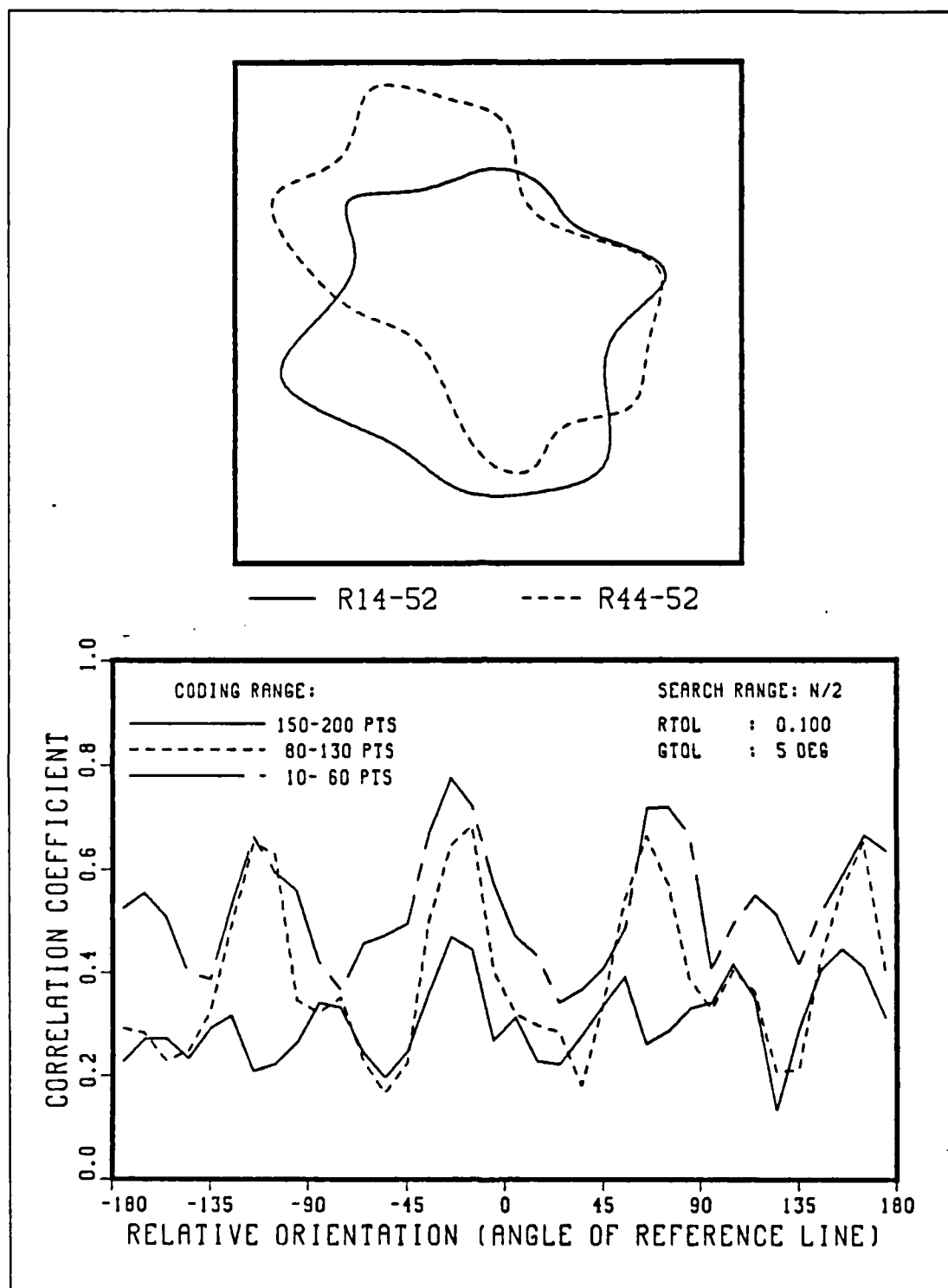


Figure 4.38 Correlation Between R14-52 and R44-52

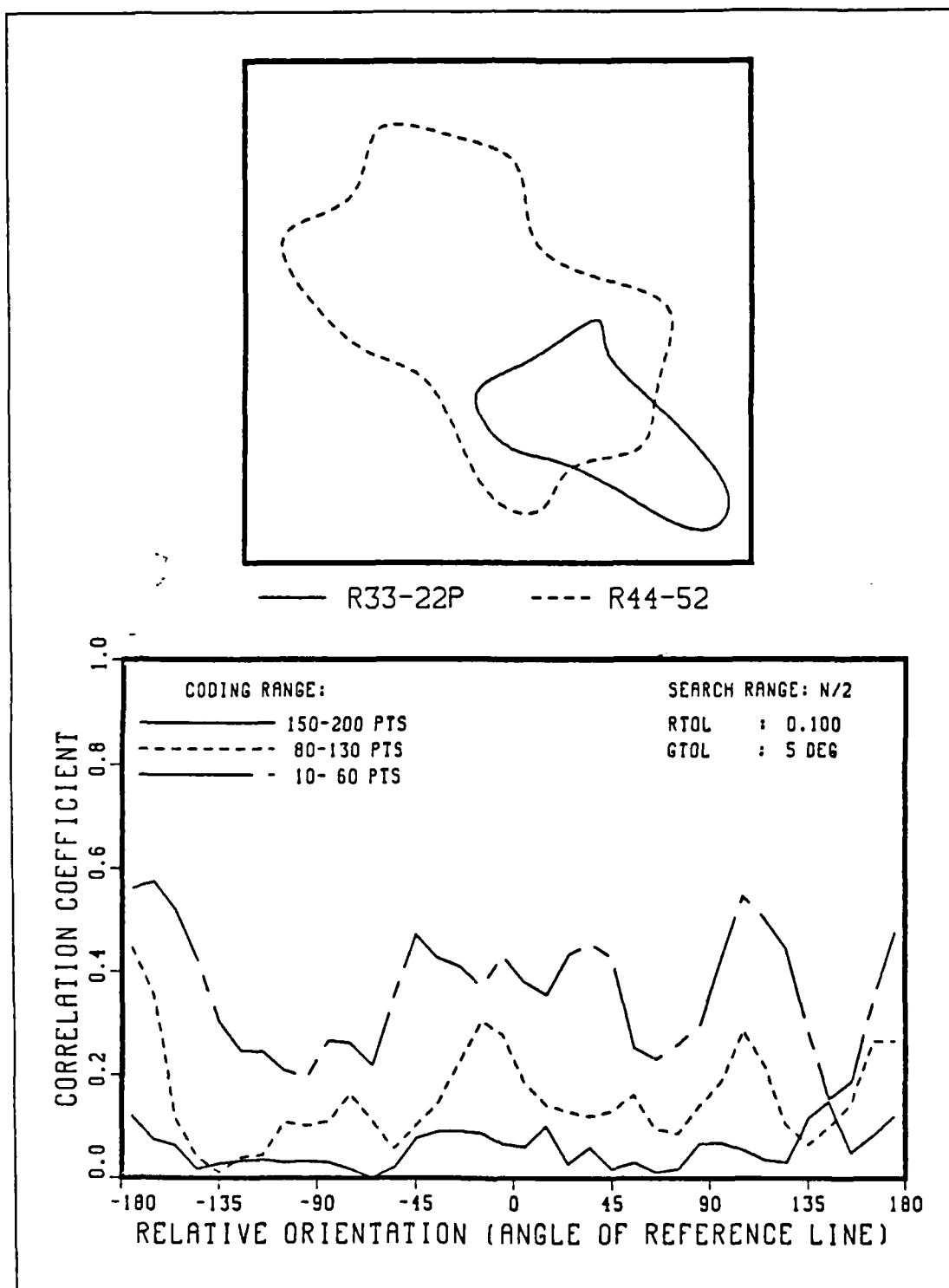
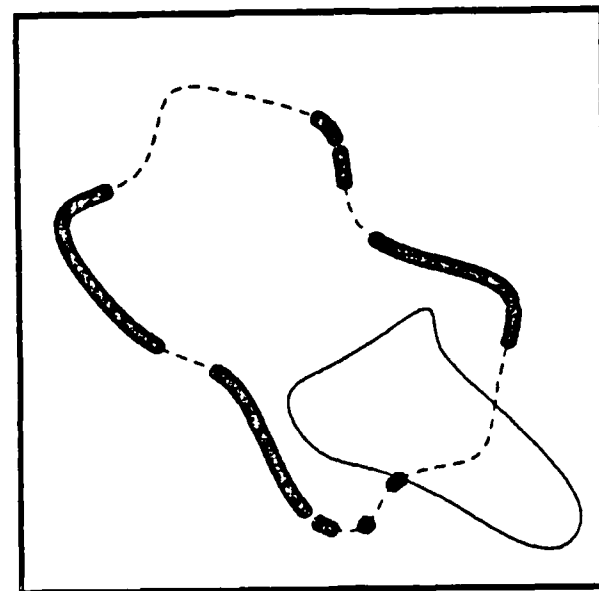
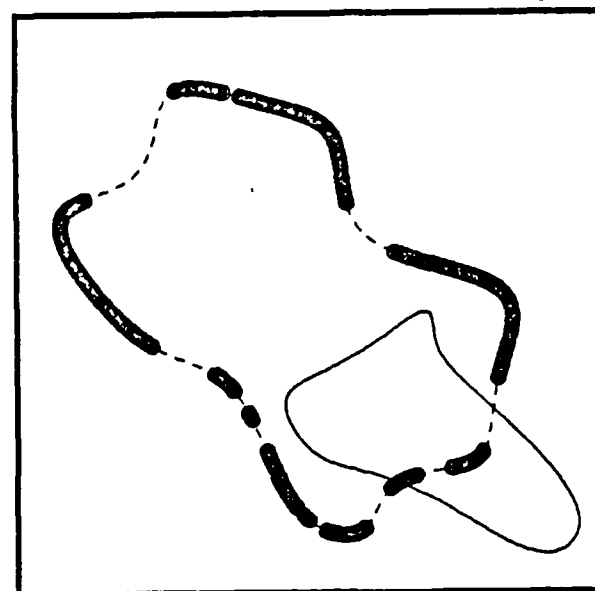


Figure 4.39 Correlation Between R33-22p and R44-52



(B) 80 TO 130 RANGE



(A) 10 TO 60 RANGE

Figure 4.40 Matched Segments Between R33-22p and R44-52
at -175 Degrees Relative Orientation

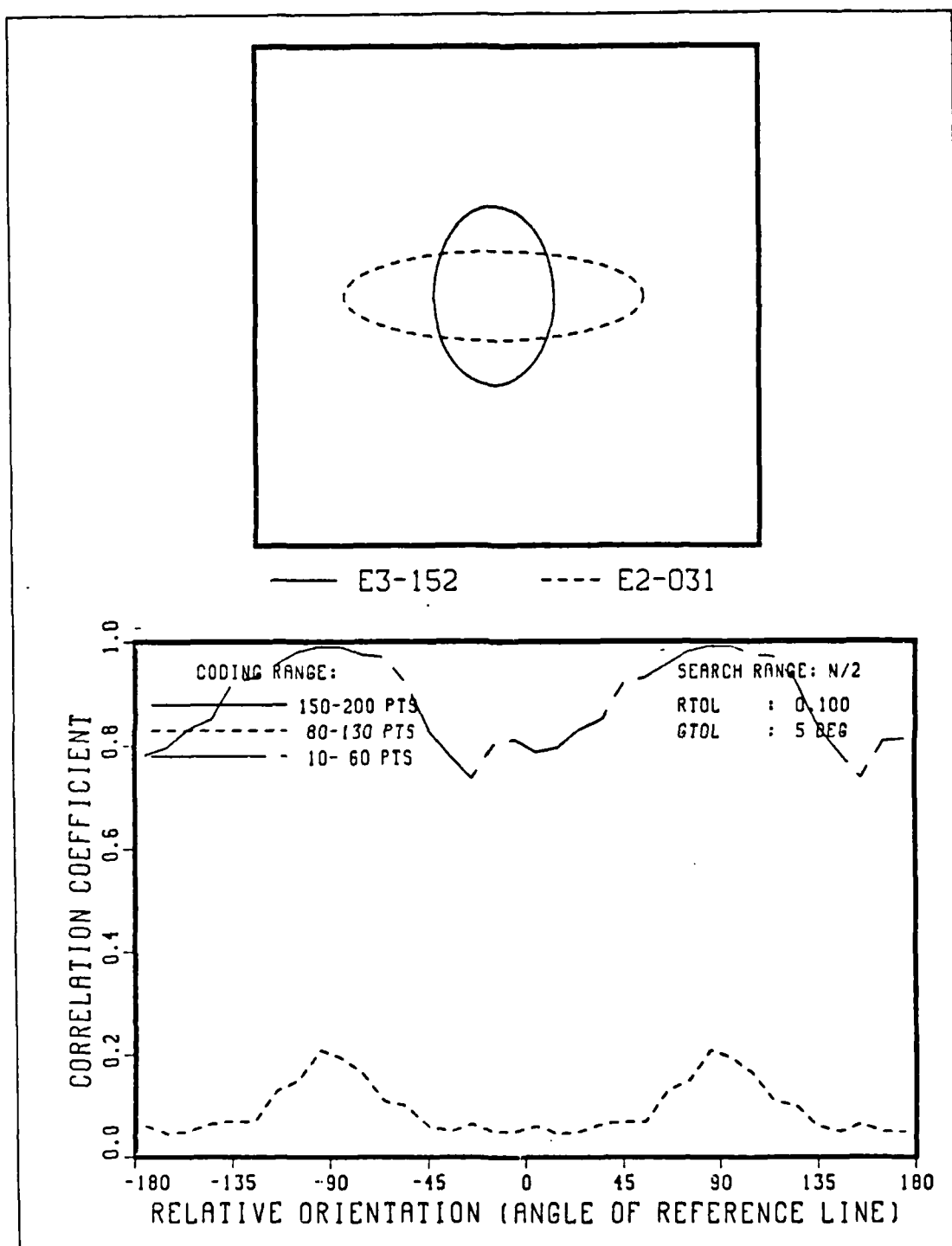


Figure 4.41 Correlation Between E3-152 and E2-031

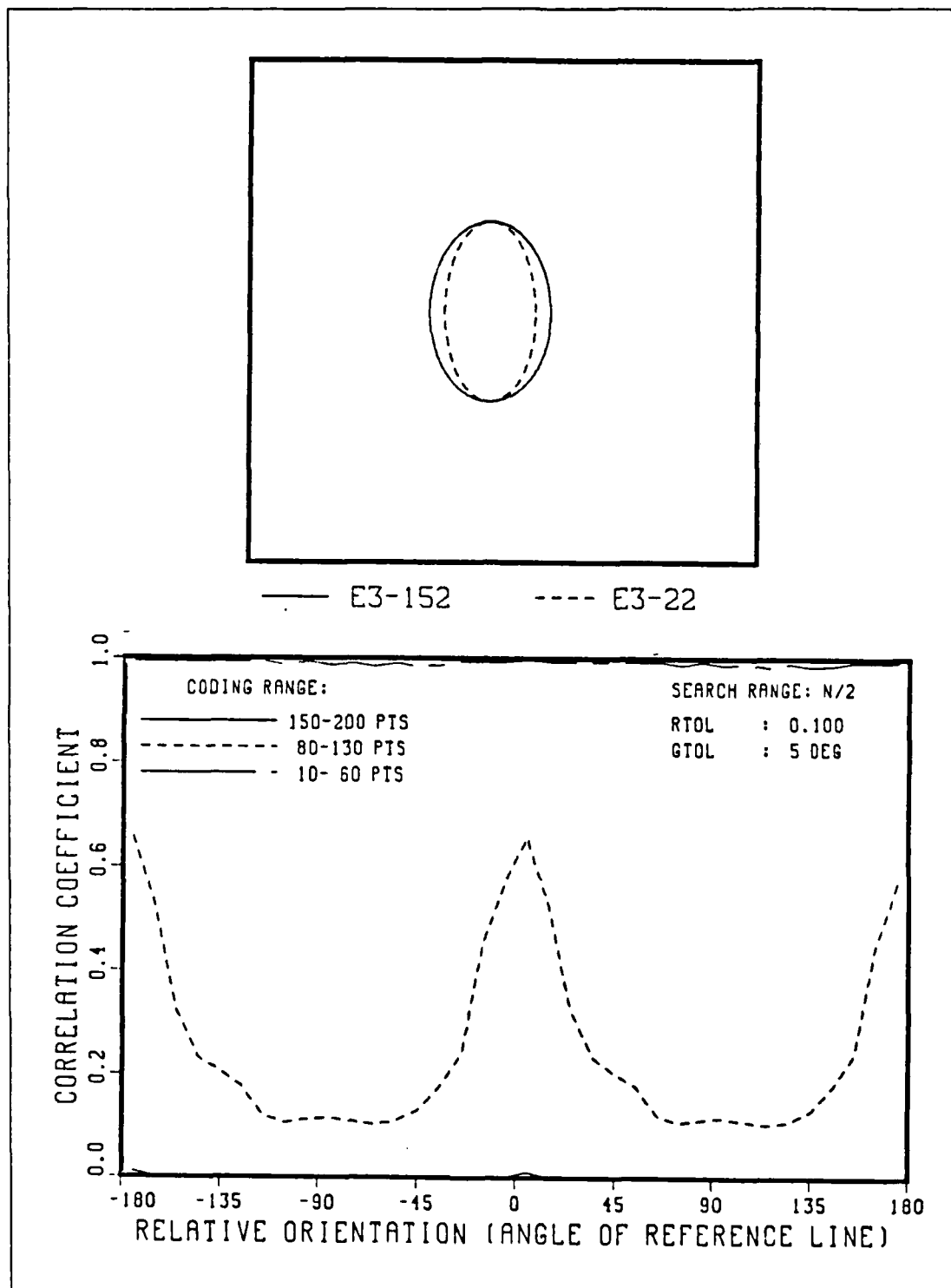


Figure 4.42 Correlation Between E3-152 and E3-22

matches ellipses of different major to minor axis ratio (b/a ratio). The b/a ratio for these ellipses are 1.5 for E3-152, 2.0 for E3-22 and 0.3 for E2-031. The results shows that ellipse of b/a ratio 1.5 is better correlated with that of ratio 2.0 than with that of ratio 0.3 (or equivalently 3.33 a/b ratio). This agrees with visual observation.

E. CONCLUSIONS

We have demonstrated the capability of this new technique and the effects of varying the various parameters on its performance. The main weakness of this technique has also been highlighted. Although the examples used have been shapes with closed boundaries, there is nothing in the algorithm that is specific to this type of shapes. The algorithm is therefore equally applicable to shapes with open boundaries.

The algorithm is implemented on the IBM 3033 computer. Computation time depends on the shapes being matched. Shapes without distinct features (or, equivalently, with lots of similar segments), such as R25-52, require the most computation. On the average, the computation of one correlation curve between two 500-points shapes takes less than 10 CPU seconds. This is with a search range of $N/2$. If this is reduced to $N/3$, this figure drops to about 6 seconds. In our examples we have used a search range of $N/2$. This is probably larger than necessary since this implies that the coding range is as large as this. One is not likely to use this large a coding range since the 'local' features in the shape being coded would then not be captured. (The choice of $N/2$ for the examples is primarily to test the ability of the algorithm to reject spurious matches from the additional checks).

AD-A168 403

RECOGNITION OF TWO-DIMENSIONAL SHAPES(U) NAVAL
POSTGRADUATE SCHOOL MONTEREY CA G P QUEK MAR 86

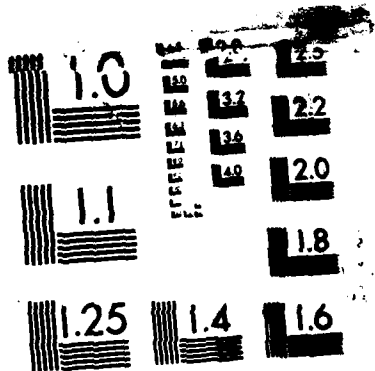
2/2

UNCLASSIFIED

F/G 9/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

V. SUMMARY

We begun with a search for a representation scheme that would be scale and orientation invariant. Such a scheme was found. However, to achieve the scale invariance, the scheme required the local behaviour of the boundary to be relatively noise free.

A more general technique was subsequently developed. The essence of this technique was the use of random boundary points in the coding, which helps to decorrelate false matches. The matching algorithm used the basic concept in Hough Transform matching but modified to remove its dependence on scale and orientation information.

This new correlation technique was applied to a large number of shapes. Results verified its ability to recognise shapes (complete or partial) of arbitrary scale and orientation and its robustness against noise. Its discrimination capability among different shapes was also demonstrated. The main weakness in the present algorithm lay in its simplistic way of summing up the correlated points without regards as to how these are distributed or interrelated.

The biggest improvement to this algorithm would come from incorporating an efficient check for consistency in the relative positions of the matched segments. The coding and matching process could also be modified to look in both 'directions', so as to reduce the 'end losses'. Further study could also be made on the choice of the various parameters used, namely the coding range, search range and tolerances on the secondary specifications. Since the reference shape would be a known entity, it would be possible, and indeed advantageous, to use different sets of parameters values for different classes of shapes, each optimised to the particular shape. In this report, we have

discussed one set of primary and secondary specifications. These may not be the most effective set available. Other possible specifications could also be examined.

Finally we note that the main contribution of this study is the suggestion of an alternative means to boundary coding, using which, an effective and efficient correlation technique could be used to match two-dimensional shapes.

APPENDIX
GENERATION OF TEST SHAPES

The shapes used for verifying the algorithm (except for ellipses) are generated using a Fourier series type method. Specifically the x,y coordinates are determined by:

$$x(\theta) = A(\theta) * \cos(\theta + \varphi)$$

$$y(\theta) = A(\theta) * \sin(\theta + \varphi)$$

with

$$A(\theta) = \exp[r(\theta)]$$

$$r(\theta) = \sum \alpha_i * \sin[f_i * \theta + \gamma_i]$$

φ = angle through which
shape is rotated

The α , γ and f can be varied to produce different shape patterns. This method ensures that the figure generated is closed. The data points would, however, not be equally spaced along the arc length. (In practice, the boundary data would be uniformly sampled). The data points are next approximated using a B-splines routine with variable knots [Ref. 18], and resampled at approximately equal arc length spacing.

There are two reasons for using B-splines. Firstly, the approximation routine available allows one to vary the closeness of fit, which enables us to introduce distortion gradually into the test shapes. Also, there has been an earlier proposal to study how the knots positions and the B-splines coefficients could be used for shape recognition purposes. (These was not carried out because of difficulties in the knots placement criteria; no satisfactory theoretical study on this has been done).

Each shape is coded with a mnemonic. Except for the ellipses, each mnemonic is prefixed with a R and has the general expression:

Rnn-nnna

where 'n' refers to a numeric and 'a' refers to an alphabet. The first numeric identifies the set of shapes (1,2,3 or 4). The second numeric indicates the number of samples (in terms of hundreds). The last numeric refers to the relative scale (1 or 2). The remaining, which may be one or two digits numeric, indicate the closeness of fit used in the spline routine. The last alphabet is optional, and indicates additional information about the shapes (p for partial, r for rotated and n for noise added). For examples,

R35-252

represents: 3 ---- shape set #3
5 ---- 500 sample points
25 ---- closeness of fit factor is 25
2 ---- relative scale of 2

and

R13-01ln

represents:
1 ---- shape set #1
3 ---- 300 sample points
01 ---- closeness of fit factor is 0.1
1 ---- relative scale of 1
n ---- noise added to portion
of the boundary

The ellipses are generated from their parametric equations. These are prefixed by the letter E. The first numeric refers to the number of sample points. The last numeric indicates the relative scale and the remaining numeric refers to the major to minor axis ratio.

LIST OF REFERENCES

1. Freeman, H., "Use of Incremental Curvature for Describing and Analysing Two-Dimensional Shapes," in *IEEE Computer Society Conference on Pattern Recognition and Image Processing*, 1979.
2. Ballard, D. H. and Brown, C. M., *Computer Vision*, Prentice-Hall, Ind., 1982.
3. Blum, H., "A Transformation for Extracting New Descriptors of Shapes," in *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn ed., MIT Press, 1967.
4. Pavlidis, T., "Algorithm for Shape Analysis of Contours and Waveforms," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-2, 1980.
5. Rosenfeld, A., "Image Analysis: Problems, Progress and Prospects," in *Pattern Recognition*, Vol 17, No 1, 1984.
6. Uhr, L., ed., *Pattern Recognition*, John Wiley and Sons, Inc., 1966.
7. Pavlidis, T., *Structural Pattern Recognition*, Springer-Verlat, 1977.
8. Wallace, T. P., Mitchell, O. R. and Fukunaga, K., "Three-Dimensional Shape Analysis using Local Shape Descriptors," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-3, 1981.
9. Davis, L. S., "Shape Matching Using Relaxation Techniques," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-1, 1979.
10. Pavlidis, T., "A Hierarchical Syntactic Shape Analysis," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-1, 1979.
11. Shapiro, L. G., "A Structural Model of Shape," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-2, 1980.
12. Duda, R. D. and Hart, P. E., *Pattern Classification and Scene Analysis*, John Wiley and Sons, Inc., 1973.

13. Gonzalez, R. C. and Wintz, P., *Digital Image Processing*, Addison-Wesley Publishing Company, 1977.
14. Faugeras, O. D., ed., *Fundamentals in Computer Vision*, Cambridge University Press, 1983.
15. Dubois, S. R. and Glanz, F. H., "An Autoregressive Model Approach to Two-Dimensional Shape Classification," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-8, 1986.
16. Ballard, D. H., "Generalizing the Hough Transform to Detect Arbitrary Shapes," in *Pattern Recognition*, Vol 13, No 2, 1981.
17. Mokhtarian, F. and Mackworth, A., "Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol PAMI-8, 1986.
18. Dierckx, P., *Algorithms for Smoothing Data with Periodic and Parametric Splines*, Report TW 55, Department of Computer Science, Katholieke Universiteit Leuven, 1981.

INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145		2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002		2
3. Department Chairman, Code 62Rr Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000		1
4. Prof. C. H. Lee, Code 62Le Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000		5
5. Prof. R. W. Hamming, Code 52Hg Department of Computer Science Naval Postgraduate School Monterey, California 93943-5000		1
6. Mr. Quek Gim Pew Blk 106, Tampines St. 11, #06-327 Singapore 1852 Republic of Singapore		3

END
DTIC

7-86